

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено

Завідувач кафедри

_____ О.В. Коваль
(підпис) (ініціали, прізвище)

“ ” _____ 2019р.

ДИПЛОМНА РОБОТА

на здобуття ступеня бакалавра

з напрямку підготовки 6.050101 “ Комп’ютерні науки ”

на тему «Моделювання забруднень повітряного середовища міста Києва »

Виконав: студент 4-го курсу, групи ТМ-52

_____ Татауров Володимир Анатолійович
(прізвище, ім’я, по батькові)

_____ (підпис)

Керівник _____ асистент Швайко В.Г.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Рецензент _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки 6.050101 “Комп’ютерні науки”

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О.В. Коваль
(підпис)

” ____ ” _____ 2018р.

ЗАВДАННЯ

на дипломну роботу студенту

Татаурову Володимиру Анатолійовичу

(прізвище, ім’я, по батькові)

1. Тема роботи «Моделювання забруднень повітряного середовища міста Києва»

керівник роботи _____ асистент. Швайко Валерій Григорович

(прізвище, ім’я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ”22” 03 2019р. №1004-с

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи мова програмування C#, платформа Microsoft Visual Studio 2015, база даних MS SQL, геоінформаційна система ArcGis

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) проаналізувати існуючі програмні засоби, розробити застосунок, спроектувати структуру бази даних, спроектувати архітектуру програми, реалізувати інтерфейс користувача системи, розробити інструкцію користувача

5. Перелік ілюстративного матеріалу

«Мета розробки», «Актуальність», «Атмосферний портрет», «Алгоритм ідентифікації», «Засоби розробки», «Концептуальна модель бази даних», «Інтерфейс користувача»

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання ” 1 ” грудня 2018 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи	01.12.2018	
2.	Вивчення та аналіз задачі	02.12.2018 – 19.12.2018	
3.	Розробка архітектури та загальної структури системи	19.12.2018 – 29.12.2018	
4.	Розробка структур окремих підсистем	09.01.2019 – 23.01.2019	
5.	Програмна реалізація системи	09.01.2019 – 19.01.2019	
6.	Оформлення пояснювальної записки	22.01.2019 – 12.03.2019	
7.	Захист програмного продукту	17.05.2019	
8.	Передзахист	31.05.2019	
9.	Захист	17.06.2019 – 21.06.2019	

Студент

(підпис)

Татауров В. А.

(прізвище та ініціали,)

Керівник роботи

(підпис)

Швайко В. Г.

(прізвище та ініціали,)

АНОТАЦІЯ

Дипломна робота присвячена розробці інструментального засобу для моделювання поширення забруднювачів в атмосфері України з використання програми моделювання та геоінформаційної системи ARCGIS. Розроблена система призначається для використання у відповідних закладах , які займаються моніторингом забруднення повітряного середовища.

Записка складається з 5 розділів, 76 сторінок, 27 рисунків, 5 таблиць, 3 додатки та 18 використаних джерел.

ABSTRACT

Diploma work is devoted to the developing of a tool for modeling the distribution of pollutants in the atmosphere of Ukraine using the simulation program and ARCGIS geoinformation system. The developed system is created for usage in the relevant institutions involved in the monitoring of air pollution.

The paper consists of 5 sections, 76 pages, 27 pictures, 5 tables, 3 additions and 18 sources.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І

ТЕРМІНІВ	7
ВСТУП.....	8
1 ПОСТАНОВКА ЗАДАЧІ.....	10
2 ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ ДЛЯ МОДЕЛЮВАННЯ	
ЗАБРУДНЕНЬ.....	12
2.1 ЕОЛ 2000	12
2.1.1 Ключові риси	13
2.2 Утіліта “Показник ризику”	16
2.3 Інтернет-додадок “Екозвіт”	16
2.4 Висновки до розділу	19
3 ЗАСОБИ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ	20
3.1 Середовище розробки Visual Studio	21
3.2 Мова програмування C#	22
3.3 Технологія .NET	25
3.4 Моделювання проекту “EcoGis” за допомогою ArcGis	28
3.5 Технологія WPF.....	29
3.6 Технологія ArcObjects SDK.....	31
3.8 SQL(Structured query language)	34
3.9 Висновки до розділу	37
4 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ МОДЕЛЮВАННЯ ЗАБРУДНЕННЯ.....	38
4.1 Архітектура системи	38
4.2 Структурна організація систем.....	43
4.3 Опис функціональних особливостей системи.....	46
4.4 Висновки до розділу	48
5 МЕТОДІКА РОБОТИ КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ	49
5.1 Інсталяція програмного забезпечення та системні вимоги.....	49
5.2 Сценарії роботи користувача з системою.....	49

5.3 Висновки до розділу	54
ВИСНОВКИ	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	56
ДОДАТОК 1	58
ДОДАТОК 2	60
ДОДАТОК 3	67

Перелік умовних позначень, символів, скорочень і термінів

ПС	Програмна система
SDK	software Development Kit — набір із засобів розробки
ПБД	Просторова база даних
ГІС	Геоінформаційна система
XML	Extensible Markup Language — Розширювана мова розмітки
LINQ	Language Integrated Query — запити, інтегровані в мову
GDI	Graphics Device Interface — інтерфейс
XAML	eXtensible Application Markup Language — декларативна мова розмітки
WPF	Windows Presentation Foundation — графічна (презентаційна) підсистема
UML	Unified Modeling Language — уніфікована мова моделювання
ООП	Об'єктно – орієнтоване програмування
ПЗ	Програмне забезпечення

ВСТУП

Забруднення повітря є одним з найважливіших питань екологічних досліджень, оскільки для нас є важливим дихати чистим повітрям, знати, які території нашої країни забруднені, та при більш детальному дослідженні цієї теми знайти шляхи вирішення проблеми забруднення атмосфери. Забруднення атмосфери виходить на перший план серед усіх екологічних проблем в зв'язку з посиленням ролі хімії в промисловому виробництві. Проблеми виживання людини вимагають конкретних відповідей на питання про зміни, що відбуваються в навколишньому середовищі. Відбуваються аварії в хімічній промисловості, що супроводжуються викидом у повітряне середовище токсичних речовин. Для вироблення правильних рішень щодо запобігання або ліквідації надзвичайних ситуацій необхідно чітко уявляти динаміку їх розвитку.

Рішення екологічних завдань проводиться на різних рівнях, в тому числі і за допомогою комп'ютерного моделювання. Математичне моделювання є найбільш перспективним напрямком вирішення завдань екології за своїми можливостями прогнозування, а також економічності матеріальних витрат і безпеки для людини проведених прогностичних експериментів. За своєю природою завдання екології та оцінки стану навколишнього середовища не допускають проведення повномасштабних натуральних експериментів, і математичне моделювання є, по суті, єдиним методом для оцінки ситуаційних ризиків, вивчення динаміки природних і техногенних катастроф та прогнозування їх наслідків, отримання загальної картини екологічної ситуації.

Однією з важливих проблем, пов'язаних з екологією, є прогнозування поширення забруднень у повітряному середовищі. До сьогодні в області математичного моделювання повітряного забруднення склалася ситуація, при якій роботи, які проводяться в світі розглядають окремі явища, але не охоплюють їх комплексно.

Модуль призначений для використання екологами для застереження

екологічних катастроф або службами порятунку для контрзаходів, якщо катастрофа все таки сталася.

При цьому важливим аспектом побудови модулю, що виконує функції взаємодії з серверною частиною є створення чіткого зрозумілого інтерфейсу з можливістю надання необхідної інформації про стан атмосфери, метеостанцій та введення даних про забруднення.

Повноцінну геоінформаційну систему неможливо створити без використання потужних сервісів для побудови карт, геокодування та отримання інформації щодо необхідної локації на мапі.

Моделювання таких складних процесів, як розповсюдження забруднень в повітрі, досить важка задача та забере багато часу на виконання її. Для вішення проблеми відображення результатів моделювання можна використати ArcGis, який являється лідером серед програмних систем для роботи з картами. Він має надійну документацію, велику базу для використання та підтримку старих продуктів, які більшість інших компаній не можуть надати.

Для інтеграції ArcGis добре підійде середовище розробки програмного забезпечення MS Visual Studio, мова програмування — C#. В купі із бібліотекою ArcObjects SDK, яку використовують для створення автономних картографічних додатків. Для розробки графічного середовища й інтерфейсу користувача використано технологію WPF. У якості бази даних використовуються MySQL.

1 ПОСТАНОВКА ЗАДАЧІ

Метою розробки є моделювання поширення забруднювачів в атмосфері України з використанням програми моделювання та геоінформаційної системи ARCGIS.

Система інтегрує дані з геоінформаційної бази даних в яку будуть включені топографічні та тематичні карти України, метеостанції, ділянок та параметри забруднення повітряного середовища.

Необхідними можливостями, які забезпечує програмне забезпечення, є:

- створити модель даних в просторовій базі даних ArcGis;
- створити програмний додаток для демонстрації метеостанцій;
- створити програмний додаток для конвертування та відображення в ArcGis результатів моделювання;
- провести аналіз наслідків розповсюдження забруднювачів.

Для моделювання для інтеграції спочатку потрібно побудувати двовимірну модель України з позначками метеостанцій і вказанням масштабу. Масштаб вказується, щоб визначити реальні розміри різних елементів на схемі. Для створення нового плану необхідно скористуватися інструментами ArcMap або завантажити готову геоінформаційну базу даних.

Інтерфейс програми включає такі елементи:

- контекстне меню;
- робочу область;
- панель елементів;
- інформаційну панель для результатів моделювання;
- інформаційну панель перегляду вхідних даних;
- діалогове вікно для виведення даних метеостанцій;
- діалогове вікно для виведення даних забруднень на ділянці.

Результатом програми має бути надання результатів моделювання забруднення атмосфери міста Києва після екстремальних ситуацій з урахуванням факторів такий як:

- балансу потоку води;
- греблі;
- водопадів;
- поздовжньої дисперсії;
- температуру;
- випаровування;
- конденсацію;
- водорості да дні.

Так як програма являється додатком до ArcMap користувач отримує доступ до всіх можливостей цієї програми такі як:

- робота з картами;
- друк карт;
- компіляція і редагування наборів геоданих;
- використання геоопрацювання для автоматизації роботи і виконання аналізу;
- організація баз геоданих і документів ArcGIS і управління ними;
- документування географічної інформації.

2 ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ ДЛЯ МОДЕЛЮВАННЯ ЗАБРУДНЕНЬ

На даний час існують програмні рішення для моделювання хімічного забруднення повітря. Кожна з них має певні переваги й недоліки. Таким чином можна взяти з них найкраще та використати в нашому продукті та спробувати не повторювати недоліки цих програм після їх аналізу.

Розглянемо деякі комп'ютерні програми, присутні на ринку ПО для аналізу довкілля. Асортимент представлений імпортованими програмами: EOL2000, Утіліта “Показник ризику”, Інтернет-додадок “Екозвіт” і т. д.

2.1. ЕОЛ 2000

Автоматизована система розрахунку розсіювання викидів шкідливих речовин в атмосферному повітрі. Призначена для оцінки впливу шкідливих викидів проєктованих та діючих (реконструюємих) підприємств на забруднення приземного шару атмосфери (Windows-версія).

Розрахункові модулі системи реалізують “Методику розрахунку концентрації в атмосферному повітрі шкідливих речовин, які містяться в викидах підприємств ОНД-86”. Система дозволяє розраховувати поля забруднення для точкової моделі джерела викиду шкідливих речовин з круглим та прямокутним вустям труби, лінійної моделі, двох моделей площинного джерела (моделі пруда-відстойника та моделі джерела, що складається із сукупності окремих точкових джерел, розміщених близько один від одного, з однаковими значеннями конструкційних та технологічних характеристик). За бажанням користувача при оцінці впливу проєктованих та реконструюємих підприємств на забруднення атмосфери розрахунок провадиться з урахуванням фонових (існуючих) концентрацій.

2.1.1 Ключові риси:

Сучасний інтерфейс(Рисунок 2.1) користувача: система реалізує стандартну для операційної системи Windows та інтуїтивно зрозумілу схему інтерфейсу користувача.

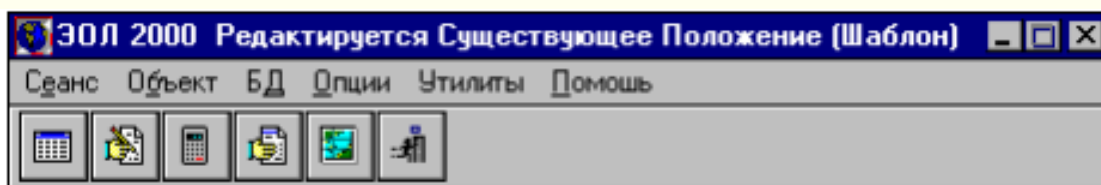


Рисунок 2.1 — демонстрація інтерфейсу

Інтегроване рішення: система зберігла традиційну для Еолу інтегровану середу, що надає можливість повністю вирішити задачу моделювання процесу розсіювання шкідливих речовин, починаючи від задачі підготовки ісходних даних та розрахунку, закінчуючи підготовкою вихідних форм та карти розсіювання шкідливих речовин на місцевості.

Інтегровані функції керування робочими директоріями (папками) забезпечує оперативність створення нових та обробку існуючих об'єктів розрахунку.

Підтримка сеансовості: користувачу надається можливість проведення багатоваріантного аналізу процесу розсіювання при різних значеннях вхідних даних. Спеціально для цієї мети розроблений механізм підтримки сеансів роботи з системою, що дозволяє ввести та зберегти історію сеансових змін.

Паперозбереження у рамках загального звіту: більш економічне розміщення вихідних форм у рамках загального звіту, зміна розмірів шрифту, що використовується для друку, а також можливість зменшення до 0 числа найбільших вкладників, що відображується на вихідній формі "Концентрації шкідливих речовин у розрахункових точках" дозволяє зменшити число друкованих аркушів на 15-30%.

Код вещества	Наименование вещества	ПДК (мг/м.куб)	Коэффициент
90	Масло минеральное	0.05000000	1.000
91	Аэрозоль сварочная	0.05000000	1.000
123	Железа окись** (в пересч	0.01000000	1.000
304	Азота окись	0.40000000	1.000
330	Ангидрид сернистый	0.50000000	1.000
337	Углерода окись	5.00000000	1.000
2704	Бензин (нефтяной, малос	5.00000000	1.000
2732	Керосин	1.20000000	1.000
2752	Чайт-спирит	1.00000000	1.000

Всего записей: 13

Вещества / Гр. суммации / Города / Промплощадки / Источники / Выбросы / Фон

Рисунок 2.2 — демонстрація роботи з базами даних

Можливість компактного збереження вихідних даних та роботу з базами даних (Рисунок 2.2): система дозволяє зберегти загальний звіт, що містить вихідні табличні матеріали та карти розсіювання, в упакованому форматі. Звіт можна продивитися та роздрукувати на робочих місцях де не інсталльований Еол 2000 за допомогою вільно розповсюджуваною утіліти "Переглядач звітів Еол 2000".

Потужний графічний процесор: об'єкти в карті розміщені на різних шарах – інсують шари вхідних даних з розміщеними на них джерелами викидів, санітарно-захисних зон та розрахунковий шар, що відображує карту розсіювання викидів шкідливих речовин. Система використовує покращений алгоритм побудови ізоліній, що дозволяє підвищувати точність відображення карти розсіювання за рахунок збільшення коефіцієнту якості побудови ізоліній. Користувач має можливість додати будь яке число користувацьких шарів з метою вводу карти-схеми ділянки місцевості на якому провадиться розрахунок розсіювання. Для цієї мети Еол 2000 реалізує універсальний графічний редактор для вводу користувацьких примитивів. Користувацькі шари чи готовий шар "Топографічна основа" можуть також бути використані з метою відображення у

вигляді підложки для побудови ізоліній готової цифрової карти місцевості, представленої у вигляді файлів формату bmp чи wmf.

Підтримка топооснови як спеціалізованого об'єкту карти Еол 2000 значно спрощує можливості виводу карти розсіювання на цифрові карти місцевості. Карта може друкуватися у заданому користувачем масштабі. Підтримка виводу карти розсіювання на графопобудівник через формування карт в форматі HPGL-2 під Windows 2000/XP забезпечує підтримку більших форматів друку. На рисунку 2.3 наведено приклад роботи програми.

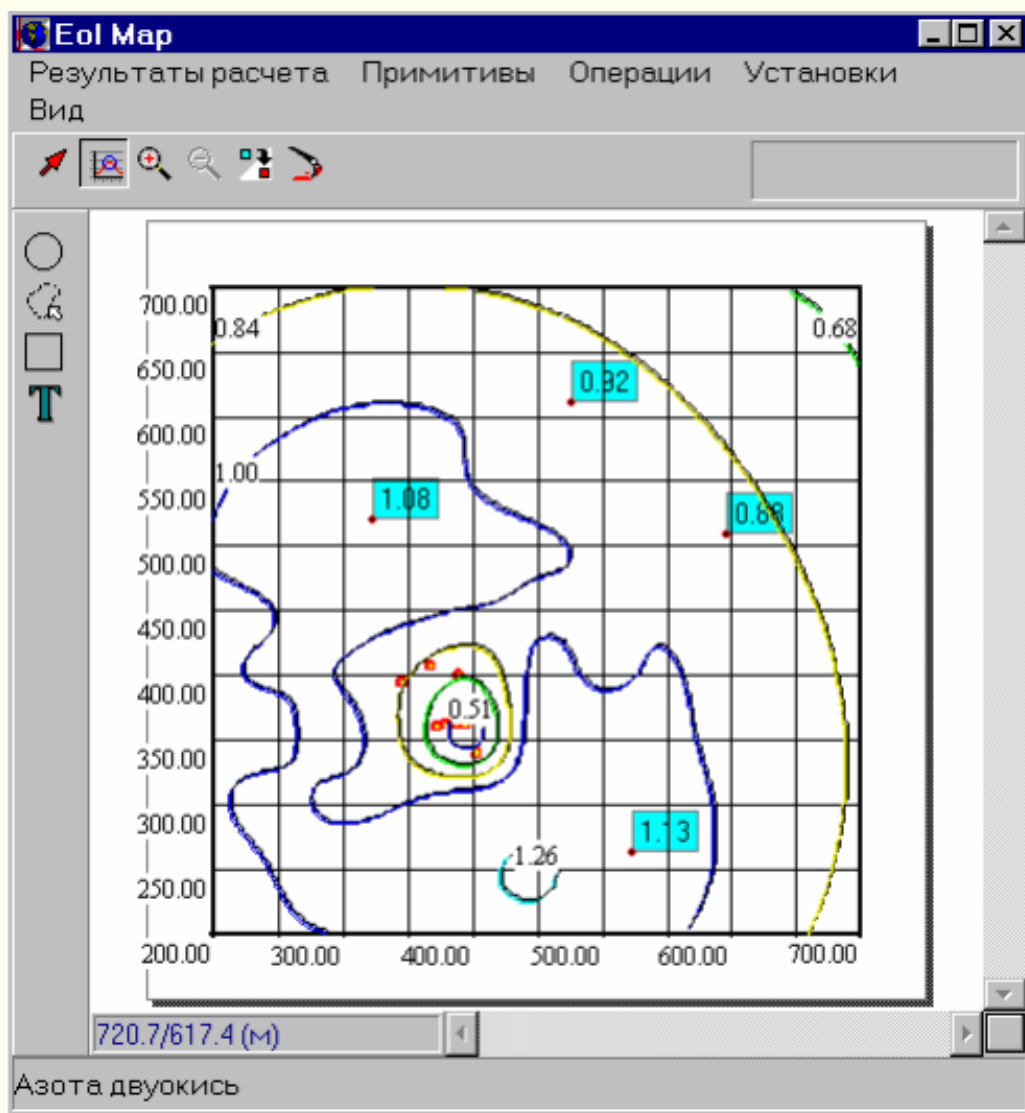


Рисунок 2.3 — моделювання забруднення

Реалізована побудова санітарно-захисної зони з урахуванням чи/та без урахування рози вітрів згідно з Державними санітарними правилами планування

та забудови населених, що дозволяє вирішувати одну з підзадач розділу "Повітряне середовище" (Склад і зміст матеріалів оцінки впливів на навколишнє середовище при проектуванні і будівництві підприємств, будинків і споруд).

Підтримується формування завдання на розрахунок "по умовчання" з урахуванням вимог Міністерства охорони навколишнього природного середовища України (реалізує уточнені вимоги до завдання розрахункових майданчиків та представлення результатів розрахунків у відповідності до "Інструкції про загальні вимоги до оформлення документів, у яких обґрунтовуються обсяги викидів, для отримання дозволу на викиди забруднюючих речовин в атмосферне повітря стаціонарними джерелами для підприємств, установ, організацій та громадян-підприємців").

У систему включені підтримка повних довідників баз даних ГДК та груп сумачії шкідливих речовин. Користувач може розширяти та доповнювати їх.

Система проста у використанні та снабджена вбудованою контекстною довідковою підсистемою із виводом повної довідки на друк.

2.2 Утіліта “Показник ризику”

Утіліта «Показник ризику» реалізує оцінку ризиків запланованої діяльності та соціальних ризиків по фактору атмосферного повітря відповідно зміні №1 «Склад і зміст матеріалів оцінки впливів на навколишнє середовище при проектуванні і будівництві підприємств, будинків і споруд».

Утіліта може експлуатуватися тільки на робочих місцях, де установлений Еол 2000 модифікації 2010 року та вище.

2.3 Інтернет-додадок “Екозвіт”

Програмний комплекс «Екозвіт» узагальнює найкращі рішення автоматизації сучасних ділових процесів з метою застосування у галузі охорони

атмосферного повітря. Застосується задля підготовки матеріалів, що обґрунтовують обсяги викидів для отримання дозволу на викиди. Поставляється у Інтернет або Інтранет варіантах.

Інтернет варіант корисний для розробки обґрунтовуючих матеріалів у середовищі віртуального офісу та з метою мінімізації витрат на підтримку власної інформаційної інфраструктури.

Інтранет редакція рекомендується для використання на ПК або у локальній мережі офісу (без доступу в Інтернет)

Обидві редакції програмного комплексу «Екозвіт» забезпечують:

— Перевірка сумісності типовим формам XML схем з формуванням звіту по несумісностям до виправлення.

— Імпорт даних з програм розрахунку забруднення, узгоджених Міністерством та експорту у Microsoft Word.

— Формування обґрунтовуючих матеріалів в друкованому та електронному вигляді. Електронний вигляд відповідає типовим формам XML схем версія 3.1.1.3. узгоджених Мінприроди України.— Актуальні довідники.

— Можливість інтеграції інформаційних матеріалів у графічному (як то генплан, карта-схема) та додатків у форматах *.doc, *.txt, *.pdf, *.epi (пояснювальні матеріали, результати розрахунків, та ін.).

Ключові переваги Інтернет-додатку «Екозвіт» :

— Доступність застосування – послуга доступна через мережу Інтернет.

— Керований доступ. Для Вашої роботи надається власний робочий простір, який доступний тільки уповноваженим Вами лицам.

— Простота застосування – забезпечується послідовним заповненням звітних даних, що не потребує високої кваліфікації користувача.

— Зменшення витрат на підтримку власної інфраструктури ПЗ. Для роботи з Інтернет-додатком достатньо мати комп'ютер, обладнаний каналом інтернет доступу со зі швидкістю 128 Кбіт/с та вище, з Microsoft Internet Explorer v 5.5 та вище та ніякої додаткової роботи для системного адміністратора.

Таким чином Ви отримуєте можливість використання новітніх нормативних, довідкових та технологічних наробок в рамках власного робочого простору в мережі Інтернет.

— Вбудована розрахункова система забезпечує автоматичне виставлення рахунків у відповідності з умовами договору на обслуговування.

— Абонентська плата за користування Інтернет-додатком викростовується Клієнтом у рамках одного календарного місяця на проведення операцій, що розраховуються на підставі діючих Тарифів у відповідності з умовами договору на обслуговування.

— Договір можна отримати по факту реєстрації. Реєстрація ведеться у відповідності з розділом «Порядок реєстрації» Інструкції користувача.

Інтернет - додаток «Екозвіт», призначений також для формування статистичної звітності у галузі охорони атмосферного повітря. Для цього забезпечуються :

— Життєвий цикл статистичної звітності через формування звітів до обробки та архівацію відпрацьованих звітів в залежності від поточної календарної дати.

— Доступність довідкової інформації - забезпечується розвинутою довідковою базою у складі Класифікаторів забруднюючих речовин та груп речовин; виробничих технологічних процесів, технологічного обладнання (установок); заходів по охороні атмосферного повітря; одиниць виміру продукції, тощо.

— Форми статистичної звітності формуються в друкованому та електронному вигляді. Електронний вигляд відповідає типовим формам XML схем, узгодженим Міністерством охорони навколишнього природного середовища України.

2.4 Висновки до розділу

Аналіз існують серйозні багатофункціональні комплекси мають досить непоганий спектр функцій як моделювання зон забруднення, наслідків хімічних аварій, забруднення повітря, прогнозу забруднення атмосфери. Але їх багатофункціональність і негативно вплинути на досвід користувача від користування програмою. Інтерфейс має бути добре продуманим і зрозумілим.

Другим недоліком цих програм являється те що програми використовують закритий формат даних їх не можна перенести в інші системи. Це веде до того, що результати цих програм важко перенести на старі версії своєї ж програми. Потрібно підтримувати документацію, що б було легко працювати з програмою.

Третій недолік цих програм являється їх ціна, щоб отримати їх потрібно заплатити за них досить велику суму.

Тому в роботі було вирішено не повторювати недоліків уже існуючих програмних рішень. Тому в наступному пункті роботи буде описано які технології і навіщо були вибрані для створення якісного програмного продукту.

3 ЗАСОБИ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ

При розробці програмного забезпечення програміст повинен вибрати такі засоби, які б полегшили його роботу, надавши всі необхідні інструменти для реалізації завдання.

При створенні програми автоматизованого проектування схем відеоспостереження було обрано такі засоби:

- середовище розробки Microsoft Visual Studio 2013;
- мову програмування C#;
- програмну технологію .NET Framework 4.0;
- технологію WPF;
- програмну систему ArcGis;

Для розробки було використано операційну систему Windows 10, основними перевагами якої є:

- стабільність роботи системи;
- зручний інтерфейс.

Також операційна система Windows 10 має точки відновлення, тому в разі збою в роботі системи можливе повне відновлення документів та інших даних. Система є багатофункціональною, при встановленні має стандартний пакет драйверів.

Для реалізації розробки інтерфейсу користувача було обрано середовище MS Visual Studio 2013, використана база геоданих, що створена в ArcGis. Для розробки графічного середовища й інтерфейсу користувача використано технологію WPF.

3.1 Середовище розробки Visual Studio

Середовище Microsoft Visual Studio (Рисунок 3.1) — лінійка продуктів компанії Microsoft, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів. Дані продукти дозволяють розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, підтримуваних Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework і Silverlight.

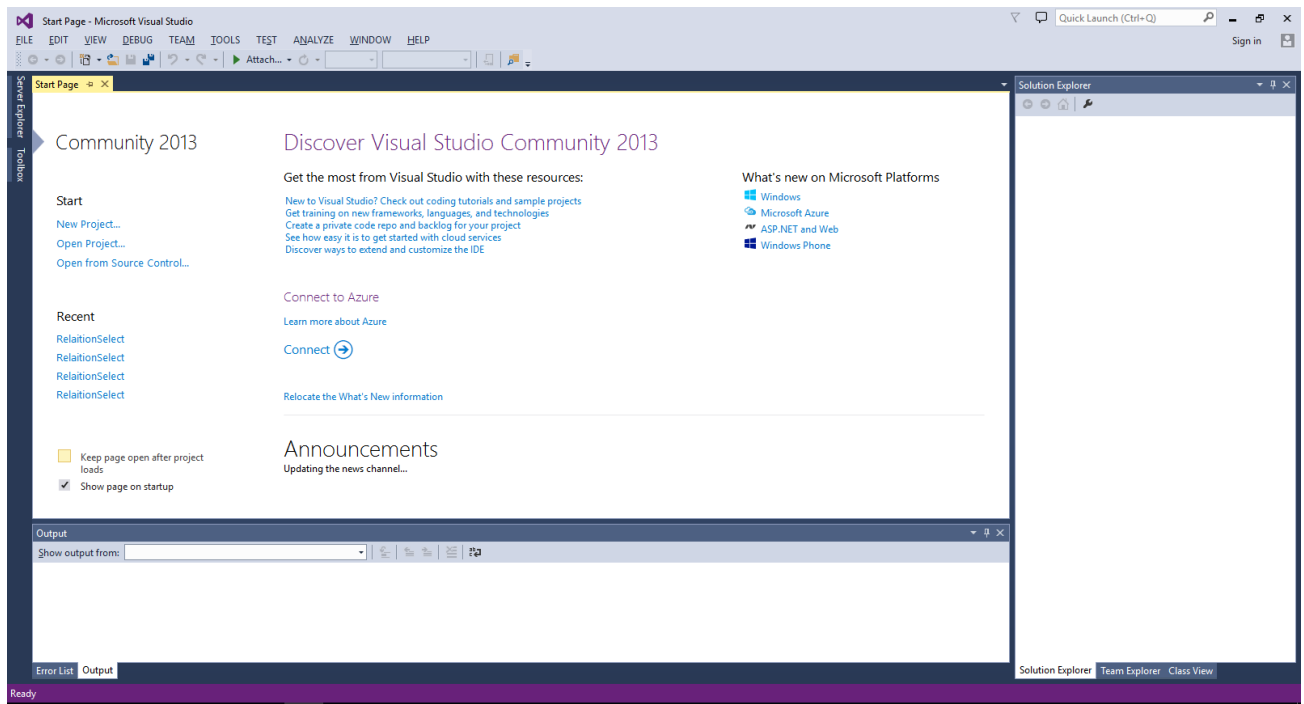


Рисунок 3.1 — Стартова сторінка Microsoft Visual Studio 2013

За допомогою цього продукту можна розробляти:

- консольні додатки;
- додатки з графічним інтерфейсом (з підтримкою технології Windows Forms);
- веб-сайти;
- веб-додатки.

Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторінга коду. Вбудований відладчик може працювати як відладчик рівня вихідного коду, так і відладчик машинного рівня. Решта вбудовуються інструменти включають в себе редактор форм для спрощення створення графічного інтерфейсу додатку, веб-редактор, дизайнер класів і дизайнер схеми бази даних. Visual Studio дозволяє створювати і підключати сторонні додатки (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду (як, наприклад, Subversion і Visual SourceSafe), додавання нових наборів інструментів (наприклад, для редагування і візуального проектування коду на предметно-орієнтованих мовах програмування) або інструментів для інших аспектів процесу розробки програмного забезпечення (наприклад, клієнт Team Explorer для роботи з Team Foundation Server).

3.2 Мова програмування C#

Мова C# [3, 4] — типізована об'єктно-орієнтована мова програмування, яка базується на сімействі мов програмування C (C, C++ , Java).

Мова C# призначена для розробки різноманітних безпечних і потужних додатків, які виконуються в середовищі .NET Framework. За допомогою мови C# (Рисунок 3.2) можна створювати звичайні додатки для Windows, XML-веб-служби, розподілені компоненти, додатки “клієнт-сервер”, додатки баз даних тощо [5].

Усі змінні і методи, включаючи метод Main, інкапсулюються в визначенні класів. Мова C# підтримує поняття інкапсуляції, наслідування і поліморфізму [6]:

— наслідуванням є можливість при описі класу вказувати на його походження (kind of relationship) від іншого класу. Наслідування дає можливість

створити новий клас, в основу якого покладено існуючий. В отриманий таким чином клас можна внести свої зміни, а потім створити нові об'єкти даного типу. Цей механізм лежить в основі створення ієрархії класів. Після абстрагування наслідування є найбільш значущою частиною загального планування системи. Похідним (derived class) є створюваний клас, похідний від базового (base class). Похідний клас наслідує всі методи базового;

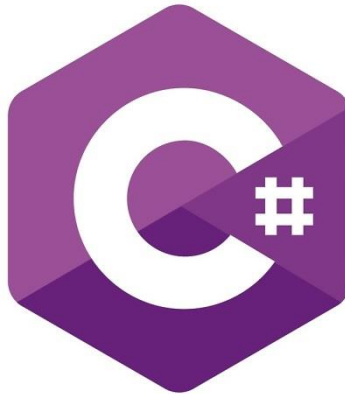


Рисунок 3.2 — логотип C#

— механізм інкапсуляції дає можливість приховати внутрішню будову об'єкта від його користувачів, надаючи через інтерфейс доступ тільки до тих членів об'єкта, з якими клієнтові можна працювати безпосередньо. Інкапсуляція має на увазі наявність межі між зовнішнім інтерфейсом класу (відкритими членами, видимими користувачам класу) і деталями його внутрішньої реалізації. Перевага інкапсуляції для розробника полягає в тому, що він може відкрити ті члени класу, які будуть залишатися статичними або незмінними, приховавши внутрішню організацію класу, більш динамічну і в більшій мірі піддається змінам. У мові C# інкапсуляція досягається шляхом призначення кожному члену класу свого модифікатора доступу — `public`, `private` або `protected`;

— властивість поліморфізму (polymorphism) дає можливість старому коду викликати новий. Завдяки поліморфізму можна змінювати поведінку класів у межах інтерфейсу, що, в свою чергу, дає можливість створювати класи, які матимуть однакове загальне призначення, проте будуть відрізнятися його реалізацією.

За допомогою поєднання цих трьох принципів досягається вся гнучкість і сила об'єктно-орієнтованого програмування. Проектування програми є процесом відображення елементів предметної області в сукупність ієрархічних об'єктів, які взаємодіють між собою так само, як це відбувається в самій предметній області.

Мова C# спрощує розробку компонентів програмного забезпечення завдяки кільком інноваційним конструкціям мови, а саме:

- інкапсульовані сигнатури методів (делегати), які підтримують безпечне повідомлення про події;
- властивості, які виступають в ролі методів доступу для закритих змінних-членів;
- атрибути з декларативними метаданими про типи під час виконання програми;
- XML-вбудовані коментарі-документації;
- функція LINQ пропонує вбудовані можливості запитів в різних джерелах даних.

Основними функціональними можливостями мови C# є [7]:

- ніяких покажчиків використовувати не потрібно. У програмах на C#, як правило, не виникає необхідність маніпулювання покажчиками;
- управління пам'яттю здійснюється автоматично за допомогою збірки сміття, тому ключове слово `delete` в C# не підтримується;
- пропонуються формальні синтаксичні конструкції для класів, інтерфейсів, структур, перерахувань і делегатів;
- надається аналогічна до C++ можливість перевантажувати операції для користувацьких типів, але без зайвих складнощів;
- підтримується програмування з використанням атрибутів;
- підтримуються анонімні методи, які дають можливість використовувати вбудовувану функцію скрізь, де потрібно використовувати тип делегата;
- підтримуються моделі “делегат-подія”;

— підтримуються лямбда-вирази, які ще більше спрощують роботу з типами делегатів в .NET.

Серіалізація в мові C# — це процес перетворення об'єкта в потік байтів (рисунок 3.3) для подальшого його зберігання в об'єкт, який можна потім передавати в пам'яті, бази даних або в файл. Основне призначення серіалізації — зберегти стан об'єкта для того, щоб мати можливість відтворити його при необхідності. Зворотний процес називається десеріалізацією.

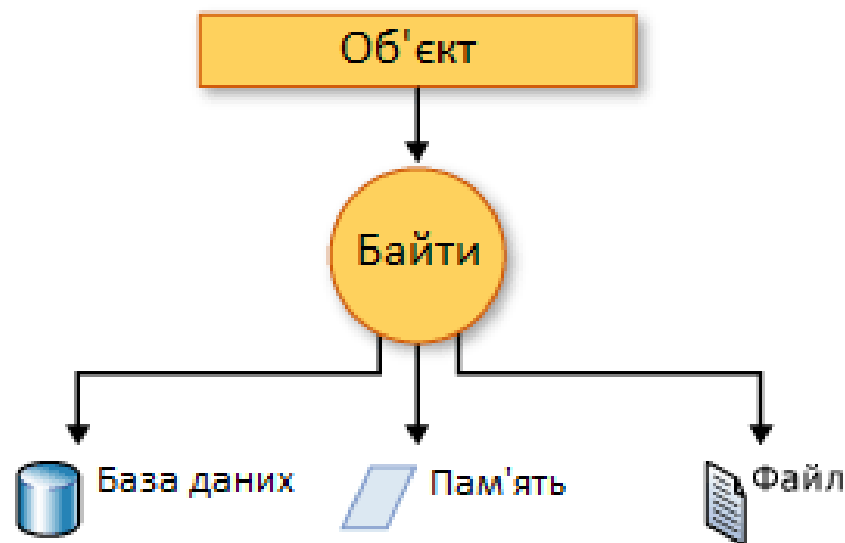


Рисунок 3.3 — Загальний процес серіалізації

3.3 Технологія .NET

Програма, написана мовою C#, виконується в середовищі .NET Framework (рисунок 3.4) [8] — інтегрованому компоненті для Windows, який містить віртуальну систему виконання (CLR-середовище), уніфікований набір бібліотек класів для всіх мов платформи .NET — BCL і бібліотеку класів FCL з розширеними бібліотеками: ASP.NET, ADO.NET, Windows Forms, WPF.

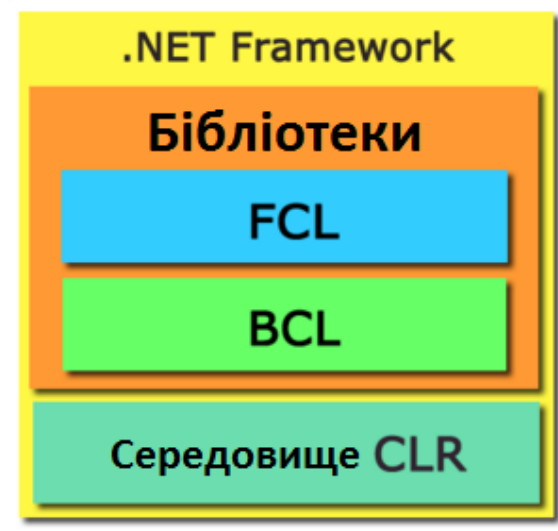


Рисунок 3.4 — Компоненти платформи .NET Framework

Середовище CLR (рисунок 3.5) є комерційною реалізацією інфраструктури CLI (common language infrastructure), міжнародного стандарту, основи середовищ виконання і розробки з тісною взаємодією мов і бібліотек.

Вихідний код, написаний мовою C #, компілюється в проміжну мову (IL) відповідно до специфікації CLI. Код IL і ресурси, такі як растрові зображення і рядки, зберігаються на диску в виконуваному файлі-збірці з розширенням EXE або DLL у більшості випадків. Збірка містить маніфест з відомостями про типи збірки, версії, мови, регіональні параметри і вимоги безпеки.

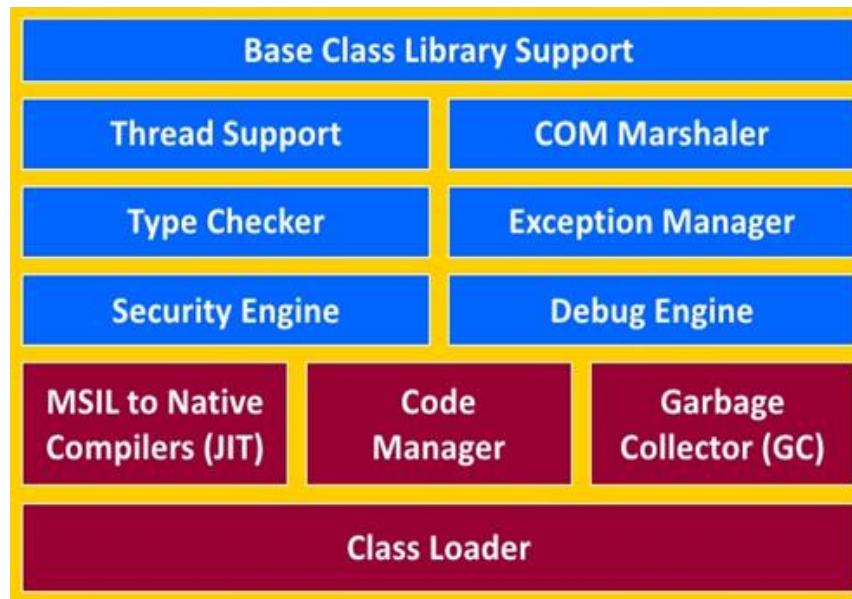


Рисунок 3.5 —Архітектура середовища CLR

При виконанні C#-програми збірка завантажується в збірку CLR залежно від відомостей в маніфесті. Далі, якщо вимоги безпеки дотримано, середовище CLR виконує JIT-компіляцію для перетворення коду IL в інструкції машинного коду. Середовище CLR також надає інші служби, які стосуються автоматичного збору сміття, обробки виключень і керування ресурсами.

Слід зазначити, що один з перших JIT-компіляторів для Java був також розроблений фірмою Microsoft (тепер в Java використовується досконаліша багаторівнева компіляція — Sun HotSpot). Сучасна технологія динамічної компіляції дозволяє досягнути аналогічного рівня швидкодії з традиційними “статичними” компіляторами (наприклад, C++) і питання швидкодії часто залежить від якості того чи іншого компілятора.

На рисунку 3.6 проілюстровано відношення під час компіляції і під час виконання між файлами з вихідним кодом C#, бібліотеками класів .NET Framework, збірками і середовищем CLR.

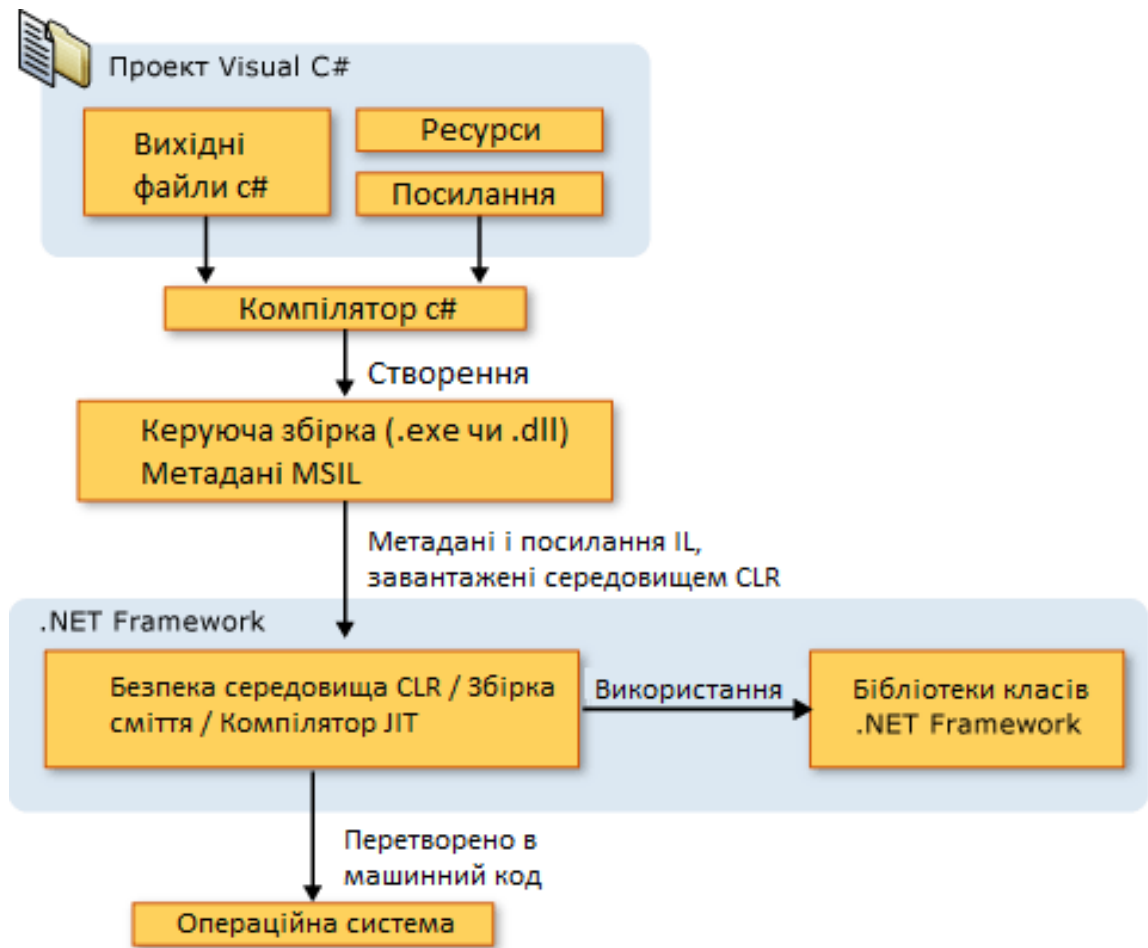


Рисунок 3.6 — Архітектура платформи .NET Framework

3.4 Моделювання проекту “EcoGis” за допомогою ArcGis

Для моделювання нашої системи ми використовуємо ArcGis(продемонстровано на рисунку 3.7). Для інтеграції ArcGis добре підійде середовище розробки програмного забезпечення MS Visual Studio, мова програмування — C#. В купі із бібліотекою ArcObjects SDK, яку використовують для створення автономних картографічних додатків. Для розробки графічного середовища й інтерфейсу користувача використано технологію WPF. У якості бази даних використовуються MySQL

ArcGIS представляє собою повну систему, яка дозволяє збирати, організовувати, управляти, аналізувати, обмінюватися і розподіляти географічну інформацію.

Платформа ArcGIS дозволяє публікувати географічну інформацію для доступу і використання будь-якими користувачами. Система доступна влюбій точці, де є можливість використання веб-браузерів, мобільних пристроїв у виді смартфонів, а також настільних комп'ютерів.

За допомогою ArcGIS усі користувачі можуть використовувати географічні знання в практичній сфері державного управління, бізнесу, науки, освіти та ЗМІ.

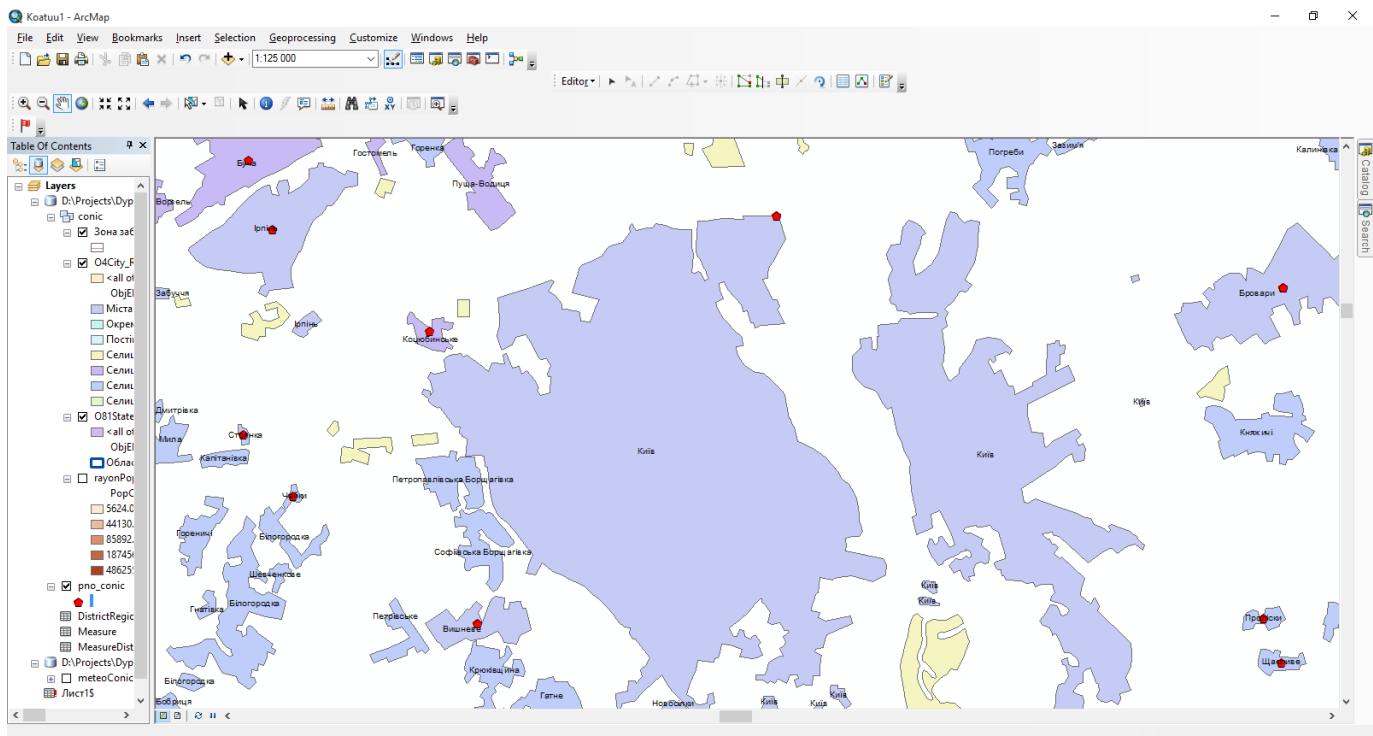


Рисунок 3.7 — Демонстрація роботи програми

3.5 Технологія WPF

Технологія Windows Presentation Foundation (або WPF) є графічною підсистемою за допомогою Microsoft для надання користувацьких інтерфейсів в додатках на базі Windows. WPF, раніше відомий як “Avalon”,

спочатку був випущений як частина .NET Framework 3.0. Замість того, щоб покладатися на старій підсистеми GDI, WPF використовує DirectX. WPF намагається забезпечити послідовну модель програмування для створення додатків і відокремлює призначений для користувача інтерфейс від бізнес-логіки. Він нагадує аналогічні моделі об'єктів XML-орієнтовані, такі як ті, які реалізовані в XUL і SVG.

WPF використовує XAML, в мові XML на основі, щоб визначити і зв'язати різні елементи інтерфейсу. Додатки WPF можуть бути розгорнуті як окремі програми для настільних або розміщених в якості впровадженого об'єкта на веб-сайті. WPF прагне об'єднати кілька загальних елементи призначеного для користувача інтерфейсу, такі як 2D/3D-рендеринг, фіксований і адаптивні документи, друкарня, векторної графіка, анімація під час виконання, і попередньо надаються засоби масової інформації. Ці елементи можуть бути пов'язані між собою і маніпулювати на основі різних подій, взаємодії користувачів і прив'язки даних.

Бібліотеки часу виконання WPF включені з усіма версіями Microsoft Windows, починаючи з Windows Vista і Windows Server 2008. Користувачі Windows XP SP2 / SP3 і Windows Server 2003 можна додатково встановити необхідні бібліотеки.

Microsoft Silverlight надає функціональні можливості, в основному підмножина WPF, щоб забезпечити вбудований веб-елементи управління, які можна порівняти з Adobe Flash. 3D-рендеринг під час виконання був підтриманий в Silverlight, так як Silverlight 5 [9].

Технологія WPF має широкий набір елементів керування (рисунок 3.8), які можна додавати на форми: текстові поля, кнопки, списки, що розкриваються, перемикачі й навіть веб-сторінки.

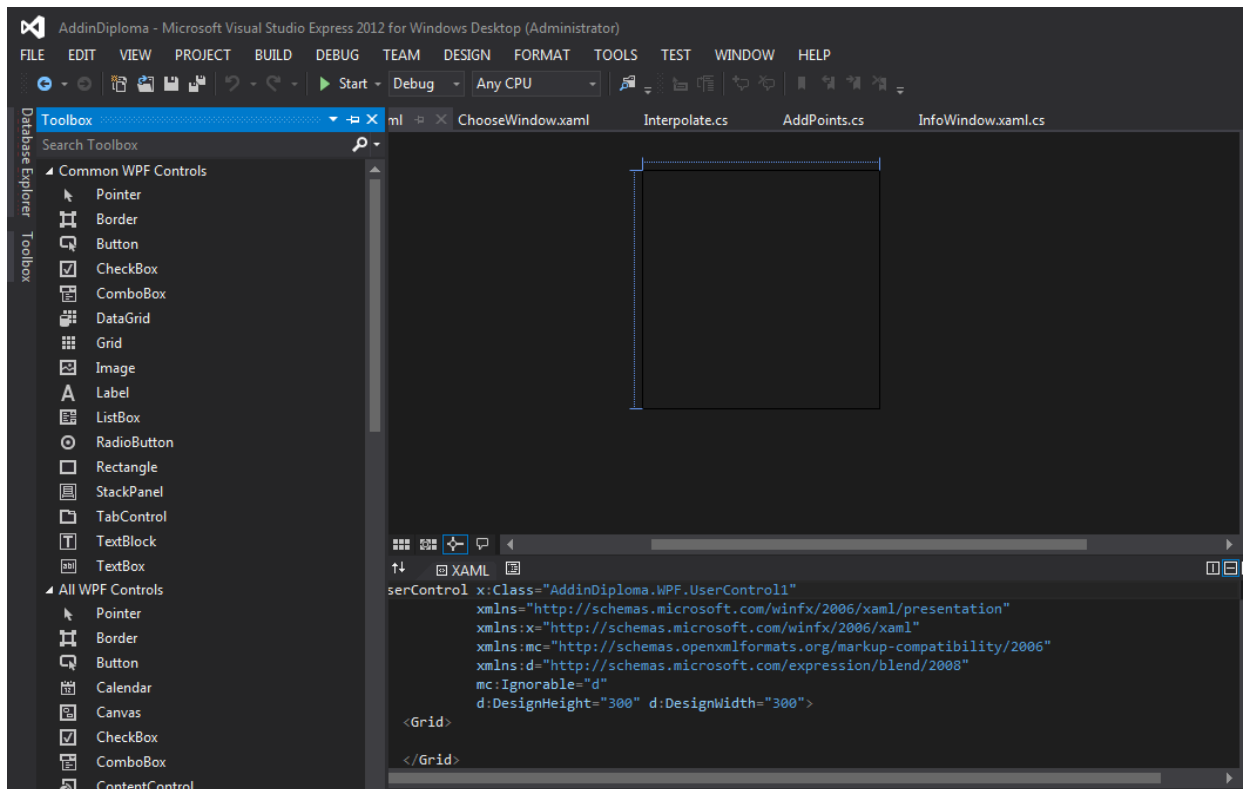


Рисунок 3.8 — Панель елементів WPF

За допомогою конструктора WPF, який підтримує перетягування, в Visual Studio можна легко створювати додатки WPF. Досить виділити елемент керування курсором і помістити його в потрібне місце на формі. Для подолання труднощів, пов'язаних з вирівнюванням елементів керування, конструктор надає такі елементи, як лінії сітки й лінії прив'язки.

3.6 Технологія ArcObject SDK

Програмні продукти компанії ESRI (США), найстарішого у світі виробника програмних засобів ГІС (фірма заснована в 1969 році).

Програмні продукти Arcgis порівняно з іншими програмами роботи з географічною інформацією мають ряд переваг:

- строга топологічність даних;
- контроль за цілісністю і топологічністю даних;

- розвинутий апарат роботи з системою координат і географічними проекціями;
- наявність розвинутого математичного апарату обробки просторових даних.

Основними її недоліками є деякі труднощі при освоєнні редактора, а також заплутана і не завжди зрозуміла ліцензійна політика і велика вартість ліцензій.

ArcObjects (Рисунок 3.9) це середовище розробки сімейства ArcGIS додатків. За допомогою Visual Basic для додатків, C Sharp (мова програмування) або Java (мова програмування) SDK для ArcGIS, що дозволяє розробникам розширювати ці додатки.



Рисунок 3.9 — логотип ArcObjects SDK

ArcObjects є бібліотекою COM-компонентів, що створюють фундамент платформи ArcGIS від ESRI. ArcObjects написана в основному на мові програмування C++. Все ArcGIS для настільних додатків засновані на ArcObjects. Оскільки ArcGIS повністю побудований на вершині ArcObjects, ви можете використовувати COM-послуг і можливостей, щоб повністю налаштувати і розширити платформу ArcGIS-meaning, що розширення моделі ArcObjects даних можна зробити легко і практично з усіма COM-сумісних мов програмування (наприклад, Visual Basic, C#, Visual Basic.NET, Java і Python). COM дозволяє компоненти для повторного використання на бінарному рівні. Іншими словами, розробники не вимагають доступу до вихідного коду ArcObjects для того, щоб розширити платформу ArcGIS. З цієї причини при

ArcObjects програміст може використовувати будь-який тип всередині системи ArcObjects, не знаючи деталей реалізації типу. Розробник повинен знати, що тип може зробити тільки. Оскільки ArcObjects заснований на стандарті COM, ви можете легко працювати з ним в поєднанні з іншими COM-об'єктів і додатків (багато програмні додатки на базі Windows, такі як Microsoft Office засновані на стандарті COM). Як уже згадувалося раніше, платформа ArcGIS була побудована з використанням типів ArcObjects (такі як класи, інтерфейси і перерахувань). У світі ArcObjects, класи використовують інтерфейси для організації властивостей і методів. Простіше кажучи, класи всередині ArcObjects використовувати тільки COM-інтерфейси, щоб виставити їх відкриті член і спілкуються один з одним. При роботі з класом ArcObjects COM, ви ніколи не працювати з властивостями і методами класу; швидше, ви завжди отримати доступ до його властивостей і методів за допомогою одного зі своїх реалізованих інтерфейсів. Як приклад, коли екземпляр об'єкта, ви можете використовувати тільки один інтерфейс. Однак, після конкретизації, ви можете запросити будь-який інший інтерфейс, який реалізується за допомогою цього об'єкта. Цей процес іноді називається інтерфейс Query (QI). Класи в ArcObjects часто мають багато інтерфейсів. Багато з компонентів ArcObjects, які складають ArcGIS використовуються у всіх трьох продуктів ArcGIS. Об'єкти в широких категоріях базових послуг, доступ до даних, аналіз карти і карт презентацій містяться у всіх трьох продуктах. Ці чотири категорії містять велику частину функціональних можливостей ГІС піддаються розробників і користувачів в ArcGIS.

Ця спільність функції серед усіх продуктів дуже важливо для розробників, щоб зрозуміти, так як це означає, що при роботі в тій чи іншій категорії, велика частина зусиль в області розвитку можуть бути передані між ArcGIS продуктів з невеликими змінами в програмному забезпеченні. Зрештою, це саме те, як розвивається архітектура ArcGIS. Повторне використання коду є основною перевагою побудови модульної архітектури, але повторне використання коду не просто приходить від створення компонентів за модульним принципом [10].

3.7 SQL(Structured query language)

Microsoft SQL Server [8] – комерційна система керування базами даних, що розповсюджується корпорацією Microsoft. Мова, що використовується для запитів — Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO щодо структурованої мови запитів SQL із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства (Рисунок 3.10).



Рисунок 3.10 – Логотип Microsoft SQL Server

Microsoft SQL Server як мову запитів використовує версію SQL, що отримала назву Transact-SQL (скорочено T-SQL), яка є реалізацією SQL-92 (стандарт ISO для SQL) з багатьма розширеннями. T-SQL дозволяє використовувати додатковий синтаксис процедур, що зберігаються і забезпечує підтримку транзакцій (взаємодія бази даних з керуючим застосунком). Microsoft SQL Server та Sybase ASE для взаємодії з мережею використовують протокол рівня застосунка під назвою Tabular Data Stream (TDS, протокол передачі табличних даних).

Microsoft SQL Server також підтримує Open Database Connectivity (ODBC)—інтерфейс взаємодії застосунків з СУБД. Версія SQL Server 2005 надає можливість підключення користувачів через веб-сервер-сервіси, що використовують протокол SOAP. Це дозволяє клієнтським програмам, не призначеним для Windows, кроссплатформенно з'єднуватися з SQL Server. Microsoft також випустила сертифікований драйвер JDBC, що дозволяє

застосункам під керування Java (таким як BEA і IBM Websphere) з'єднуватися з Microsoft SQL Server 2000 і 2005.

SQL Server підтримує дзеркалювання та кластеризацію баз даних [8]. Кластер серверу SQL—це сукупність однаково конфігурованих серверів; така схема допомагає розподілити робоче навантаження між декількома серверами. Усі сервери мають одне віртуальне ім'я, а дані розподіляються за IP-адресами машин кластеру протягом робочого циклу. Також у разі відмови або збою на одному з серверів кластеру доступне автоматичне перенесення навантаження на інший сервер.

SQL Server 2005 має вбудовану підтримку .NET Framework. Завдяки цьому, процедури бази даних, що зберігаються, можуть бути написані на будь-якій мові платформи .NET з використанням повного набору бібліотек, доступних для .NET Framework. На відміну від інших процесів, .NET Framework виділяє додаткову пам'ять і будує засоби керування SQL Server, не використовуючи вбудовані засоби Windows. Це підвищує продуктивність порівняно із загальними алгоритмами Windows, оскільки алгоритми розподілу ресурсів спеціально налагоджені для використання у структурах SQL Server.

MS SQL Server — компактний багатопотоковий сервер баз даних. Характеризується високою швидкістю, стійкістю і простотою використання.

MS SQL Server вважається гарним рішенням для малих і середніх застосувань. Сирцеві коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в UNIX-системах, де є підтримка багатопоточності, що підвищує продуктивність системи в цілому.

Можливості сервера MS SQL Server:

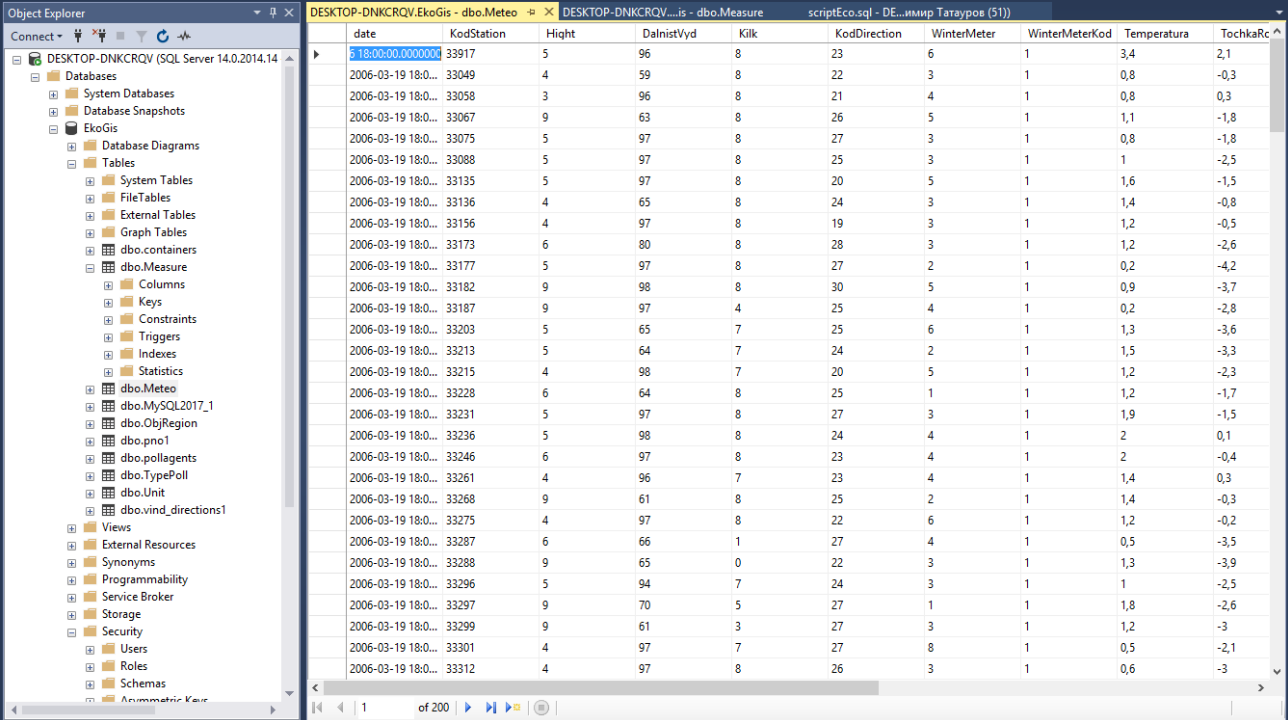
- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн;
- висока швидкість виконання команд;

- наявність простої і ефективної системи безпеки.

Серед основних переваг MS SQL Server відзначають наступні:

- Масштабованість. MS SQL Server може підтримувати роботу БД значних розмірів, що підтверджують її реалізації в Yahoo!, Google, HP, Associated Press.
- Переносність. MS SQL Server працює на різних платформах, серед яких Unix, Linux, Windows, OS/2, Solaris, Mac OS. Окрім того, MS SQL Server працює на різних платформах.
- Зв'язаність. MS SQL Server має мережеву структуру. До MS SQL Server можна одержувати доступ із будь-якої точки Internet кільком користувачам одночасно. MS SQL Server має цілий ряд програмних інтерфейсів додатків (Application Programming Interface –API), які дозволяють встановлювати з'єднання з MS SQL Server із додатків, написаних на таких мовах як C, C++, C#, Perl, PHP, Java, Python.
- Безпека. MS SQL Server має систему контролю доступу до даних, забезпечує шифрування даних при передаванні.
- Швидкість функціонування.
- Зручність експлуатації. MS SQL Server досить зручно встановлюється та реалізується, легко адмініструється.

SQL Server Management Studio (Рисунок 3.11) – утиліта компанії Microsoft для Microsoft SQL Server для конфігурування, керування та адміністрування усіх компонентів Microsoft SQL Server. Утиліта включає в себе редактор скриптів та графічну програму, котра працює з об'єктами та налаштуваннями серверу. Головним інструментом SQL Server Management Studio являється Object Explorer, який дозволяє користувачу проглядати, витягувати об'єкти сервера, а також повністю керувати ними.



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane displays the database structure for 'DESKTOP-DNKCRCV (SQL Server 14.0.2014.14)'. The database contains several tables, including 'dbo.Meteo', 'dbo.MySQL2017_1', 'dbo.ObjRegion', 'dbo.pno1', 'dbo.pollagents', 'dbo.TypePoll', 'dbo.Unit', 'dbo.vind_directions1', 'dbo.containers', 'dbo.tables', 'dbo.measure', 'dbo.statistics', 'dbo.constraints', 'dbo.triggers', 'dbo.indexes', 'dbo.views', 'dbo.external_resources', 'dbo.synonyms', 'dbo.programmability', 'dbo.service_broker', 'dbo.storage', 'dbo.security', 'dbo.users', 'dbo.roles', 'dbo.schemas', and 'dbo.asymmetric_keys'. The main pane displays a query result for the 'dbo.Meteo' table. The query is 'select * from dbo.Meteo'. The result set contains 200 rows of data, with columns: 'date', 'KodStation', 'Hight', 'DalnistVyd', 'Kilk', 'KodDirection', 'WinterMeter', 'WinterMeterKod', 'Temperatura', and 'TochkaRc'. The data shows measurements for various stations over time, with values ranging from 0 to 200 for 'KodStation' and 'Hight', and from -3.5 to 2.1 for 'TochkaRc'.

date	KodStation	Hight	DalnistVyd	Kilk	KodDirection	WinterMeter	WinterMeterKod	Temperatura	TochkaRc
2006-03-19 18:00:00.0000000	33917	5	96	8	23	6	1	3,4	2,1
2006-03-19 18:00:00.0000000	33049	4	59	8	22	3	1	0,8	-0,3
2006-03-19 18:00:00.0000000	33058	3	96	8	21	4	1	0,8	0,3
2006-03-19 18:00:00.0000000	33067	9	63	8	26	5	1	1,1	-1,8
2006-03-19 18:00:00.0000000	33075	5	97	8	27	3	1	0,8	-1,8
2006-03-19 18:00:00.0000000	33088	5	97	8	25	3	1	1	-2,5
2006-03-19 18:00:00.0000000	33135	5	97	8	20	5	1	1,6	-1,5
2006-03-19 18:00:00.0000000	33136	4	65	8	24	3	1	1,4	-0,8
2006-03-19 18:00:00.0000000	33156	4	97	8	19	3	1	1,2	-0,5
2006-03-19 18:00:00.0000000	33173	6	80	8	28	3	1	1,2	-2,6
2006-03-19 18:00:00.0000000	33177	5	97	8	27	2	1	0,2	-4,2
2006-03-19 18:00:00.0000000	33182	9	98	8	30	5	1	0,9	-3,7
2006-03-19 18:00:00.0000000	33187	9	97	4	25	4	1	0,2	-2,8
2006-03-19 18:00:00.0000000	33203	5	65	7	25	6	1	1,3	-3,6
2006-03-19 18:00:00.0000000	33213	5	64	7	24	2	1	1,5	-3,3
2006-03-19 18:00:00.0000000	33215	4	98	7	20	5	1	1,2	-2,3
2006-03-19 18:00:00.0000000	33228	6	64	8	25	1	1	1,2	-1,7
2006-03-19 18:00:00.0000000	33231	5	97	8	27	3	1	1,9	-1,5
2006-03-19 18:00:00.0000000	33236	5	98	8	24	4	1	2	0,1
2006-03-19 18:00:00.0000000	33246	6	97	8	23	4	1	2	-0,4
2006-03-19 18:00:00.0000000	33261	4	96	7	23	4	1	1,4	0,3
2006-03-19 18:00:00.0000000	33268	9	61	8	25	2	1	1,4	-0,3
2006-03-19 18:00:00.0000000	33275	4	97	8	22	6	1	1,2	-0,2
2006-03-19 18:00:00.0000000	33287	6	66	1	27	4	1	0,5	-3,5
2006-03-19 18:00:00.0000000	33288	9	65	0	22	3	1	1,3	-3,9
2006-03-19 18:00:00.0000000	33296	5	94	7	24	3	1	1	-2,5
2006-03-19 18:00:00.0000000	33297	9	70	5	27	1	1	1,8	-2,6
2006-03-19 18:00:00.0000000	33299	9	61	3	27	3	1	1,2	-3
2006-03-19 18:00:00.0000000	33301	4	97	7	27	8	1	0,5	-2,1
2006-03-19 18:00:00.0000000	33312	4	97	8	26	3	1	0,6	-3

Рисунок 3.11 – Приклад бази даних нашого проекту за допомогою SQL Server Management Studio

3.8 Висновки до розділу

Отже після проведення аналізу вибраних методів та засобів розробки програмного продукту. Можна сказати, що вибраний стек технологій та засобів добре підійде для виконання поставленої задачі.

Для моделювання нашої системи ми використовуємо ArcGis. Для інтеграції ArcGis добре підійде середовище розробки програмного забезпечення MS Visual Studio, мова програмування — C#. В купі із бібліотекою ArcObjects SDK, яку використовують для створення автономних картографічних додатків. Для розробки графічного середовища й інтерфейсу користувача використано технологію WPF. У якості бази даних використовуються MySQL

4 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ МОДЕЛЮВАННЯ ЗАБРУДНЕННЯ

Створена програмна система складається з модулів, які працюють з однієї просторовою базою даних.

4.1 Архітектура системи

Розроблена система є клієнтським додатком. Вона включає в себе такі логічні модулі (рисунок 4.1):

- модуль інтерфейса користувача;
- модуль модель;
- модуль контролер.

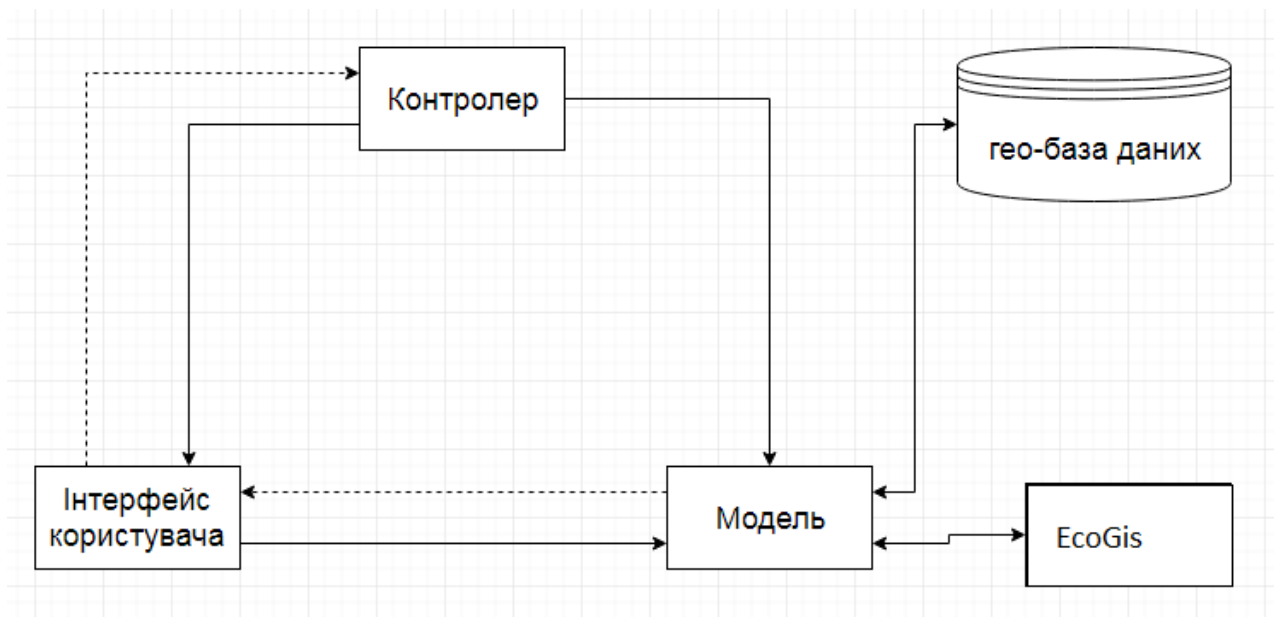


Рисунок 4.1 — Архітектура програми

Модуль модель надає дані і методи роботи з ними: запити до гео-бази даних, перевірка на коректність даних, генерацію файлів для ЕcoGis та розшифрування отриманого результату. Модель не залежить від уявлення (не

знає як візуалізувати дані) і контролера (не має точок взаємодії з користувачем) просто надаючи доступ до даних і управління ними.

Модуль інтерфейсу користувача відповідає за отримання необхідних даних з моделі і відправляє їх користувачеві.

Модуль контролер забезпечує «зв'язок» між користувачем і системою. Контролює і направляє дані від користувача до системи і навпаки.

Просторова база даних яка використовується в проєкті складається з багатьох таблиць, далі ми розберемо основні таблиці(Рисунок 4.2).

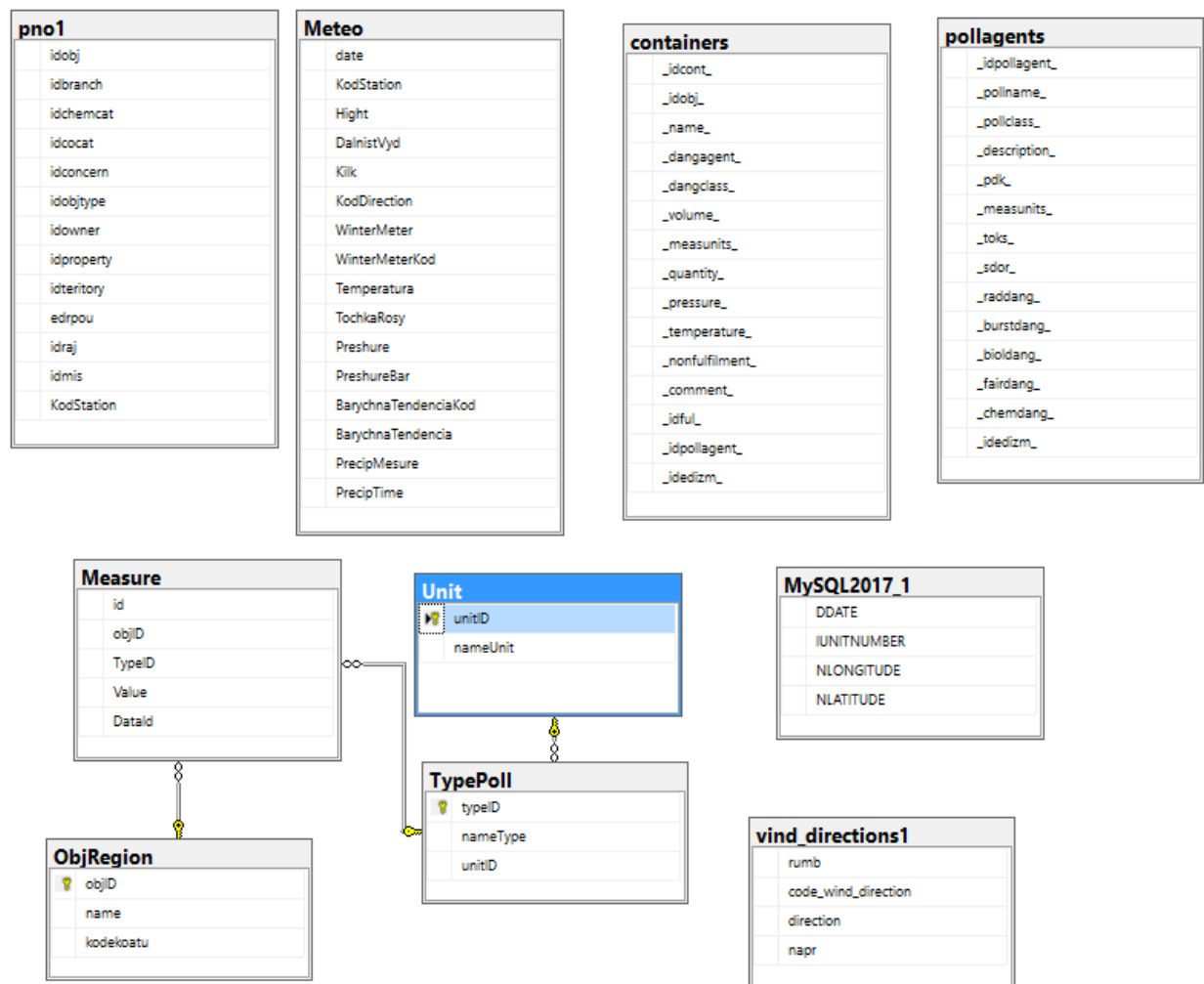


Рисунок 4.2 — Основні таблиці

Однією з основних таблиць являється таблиця Meteo, в якій знаходиться основна інформація про метеостанції, температуру повітря, висота, назва та різні коефіцієнти (рисунок 4.3).

Meteo	
date	
KodStation	
Hight	
DalnistVyd	
Kilk	
KodDirection	
WinterMeter	
WinterMeterKod	
Temperatura	
TochkaRosy	
Preshure	
PreshureBar	
BarychnaTendenciaKod	
BarychnaTendencia	
PrecipMesure	
PrecipTime	

Рисунок 4.3 — Таблица метеостанцій

Другою таблицею являється Pollagents в якій збережені дані по хімічному забрудненню в певних точках, які були зроблені відповідні заміри (рисунок 4.4).

pollagents	
idpollagent	
pollname	
pollclass	
description	
pdk	
measunits	
toks	
sdor	
raddang	
burstdang	
bioldang	
fairdang	
chemdang	
idedizm	

Рисунок 4.4 — Таблица ділянок

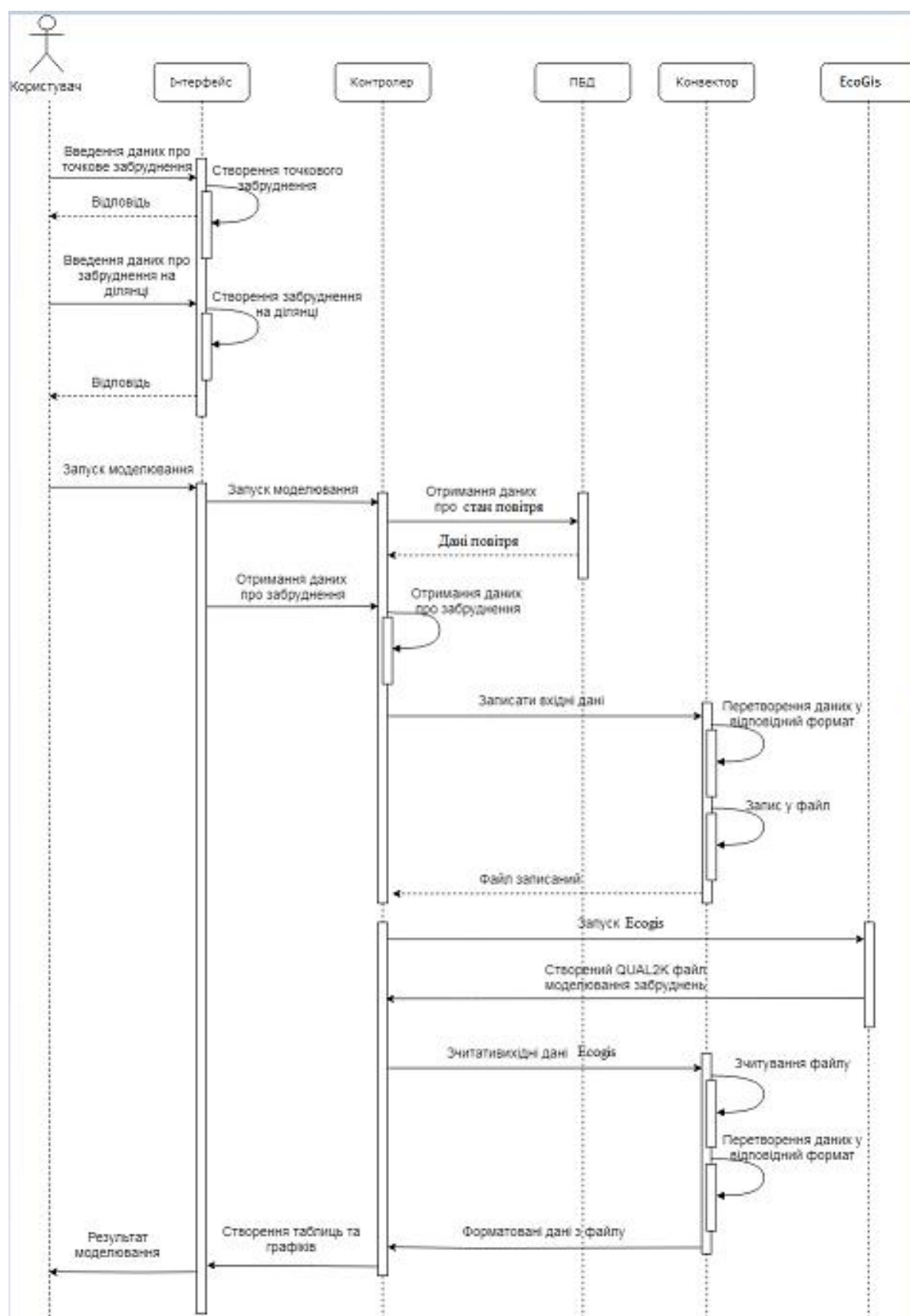


Рисунок 4.5 — Діаграма послідовності користування програмою

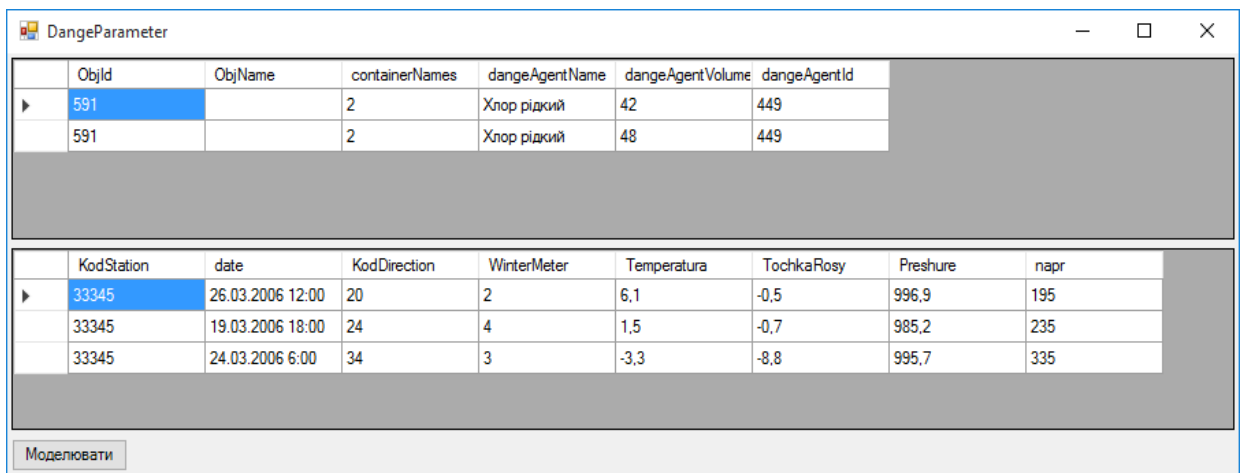
З діаграми послідовності (рисунок 4.5) можна розібратися, як взаємодіють об'єкти між собою за плином часу. Та зрозуміти, як в систему проводиться інтеграція з Eсogis. Цій програмі на вхід потрібно дати файл в форматі .q2k, з всіма даними про річку та її ділянки, забруднення та коефіцієнти розрахунків, по якому він і проведе моделювання забруднень і створе новий файл результатів в форматі .out. В ньому будуть створені таблиці результатів моделювання.

Основною проблемою було те, що модуль Eсogis написаний на мові програмування C# і тому число з рухомою комою там представлені в такому вигляді як .09 а не 0,09 як виводить наприклад Fortran тому потрібно, було створити модуль конвектор, що буде перетворювати різні типи даних в ті які буде розуміти EсoGis.

Друга проблема, це зчитування даних з файлу результатів моделювання та створення відповідних таблиць. Формат виведення видно на рисунку 4.6. Для цього був створений алгоритм, який буде і створювати таблиці в ПБД.

Назву таблиці він бере з інформації, яка знаходилась між знаками **, після цього він зчитував назви колонок з наступного рядка, але створювалася таблиця вже після того як зчитає самі дані таблиці, щоб визначити формат цих колонок.

Після всіх цих операцій створюється таблиця в ПБД з відповідною назвою, колонками та даними. І вже з цими таблицями проводяться основні дії для виведення результатів, а саме створення графіків та відображення на карті.



ObjId	ObjName	containerNames	dangeAgentName	dangeAgentVolume	dangeAgentId
591		2	Хлор рідкий	42	449
591		2	Хлор рідкий	48	449

KodStation	date	KodDirection	WinterMeter	Temperatura	TochkaRosy	Preshure	napr
33345	26.03.2006 12:00	20	2	6,1	-0,5	996,9	195
33345	19.03.2006 18:00	24	4	1,5	-0,7	985,2	235
33345	24.03.2006 6:00	34	3	-3,3	-8,8	995,7	335

Моделювати

Рисунок 4.6 — Результат роботи Eсogis

4.2 Структурна організація систем

Для реалізації функціональних можливостей системи розроблено спеціальні класи (рисунок 4.7).

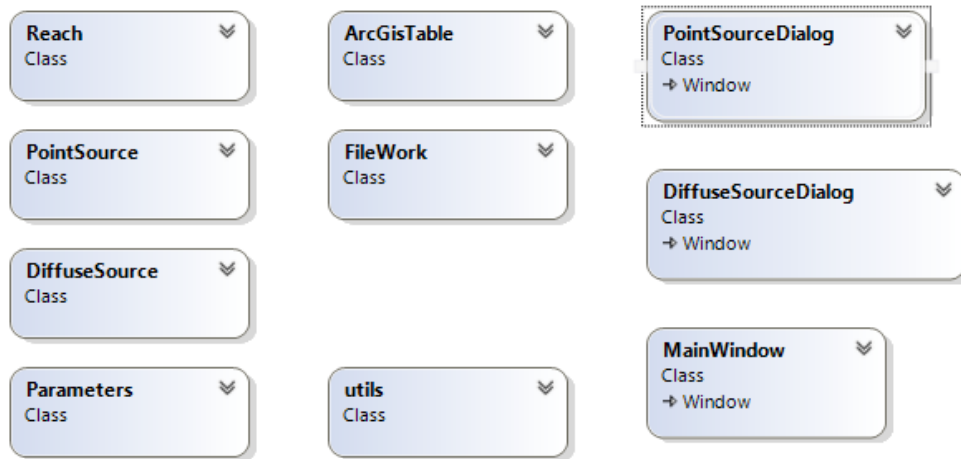


Рисунок 4.7 — Загальна діаграма класів системи

Клас MainClass.cs є класом, що відповідає за реагування на натиск кнопки в ArcMap середовищі. Та відкриває вікно для користування нашим додатком (рисунок 4.8).

Клас MainWindow.cs є класом, що відповідає за відображення та введення всіх даних, які потрібні для моделювання забруднень зоні атмосфери. Тут виводиться інформація про метеостанції, які можна вибрати для моделювання. По цій інформації вибираються точки річки на яких було проведені відповідні заміри та виводяться дані цих замірів на відповідну вкладку де можна вести нові точки, редагувати старі, або навіть видалити.

Виводяться та вводяться параметри розрахунків, такі як за якою формулою розраховувати забруднення повітря .

А також сам результат моделювання у вигляді таблиць та графіків.

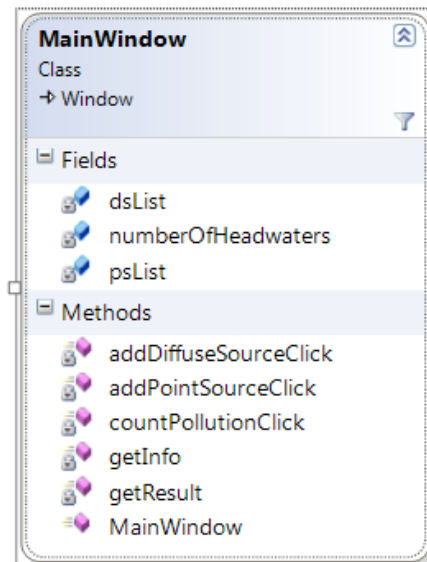


Рисунок 4.8 — Клас MainWindow

Клас **PointSourceDialog.cs** відповідає за введення даних про точкове забруднення.

Клас **DiffuseSourceDialog.cs** відповідає за введення даних про забруднення повітря.

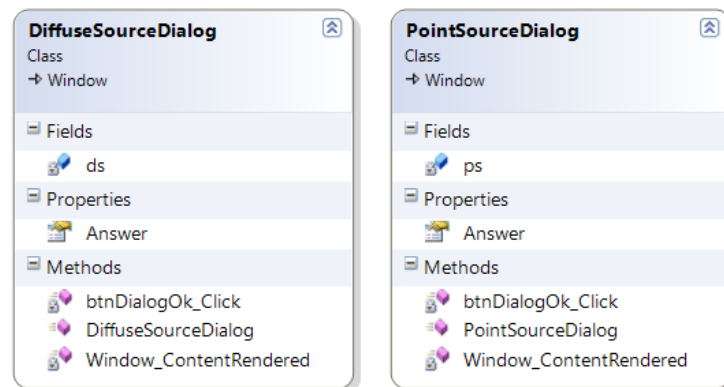


Рисунок 4.9 — Класи PointSourceDialog, DiffuseSourceDialog

Клас **Reach.cs** відповідає за взаємодію з метеостанціями на карті, а саме додавання, редагування, видалення та вибрати записи із гео-бази даних. Перетворює їх на ділянки для програми **Ecogis**.

Клас `DiffuseSource.cs` відповідає за взаємодію із забрудненнями на ділянці, а саме додавання, редагування, видалення та вибрати записи із гео-бази даних. Перетворює їх у відповідні дані для програми `Ecogis`.

Клас `PointSource.cs` відповідає за взаємодію із точковими забрудненнями, а саме додавання, редагування, видалення та вибрати записи із гео-бази даних. Перетворює їх у відповідні дані для програми `Ecogis`.

Клас `Parameters.cs` відповідає за всі параметри та коефіцієнти розрахунку, додавання, редагування, видалення та вибрати записи із гео-бази даних. Перетворює їх у відповідні дані для програми `Ecogis`.

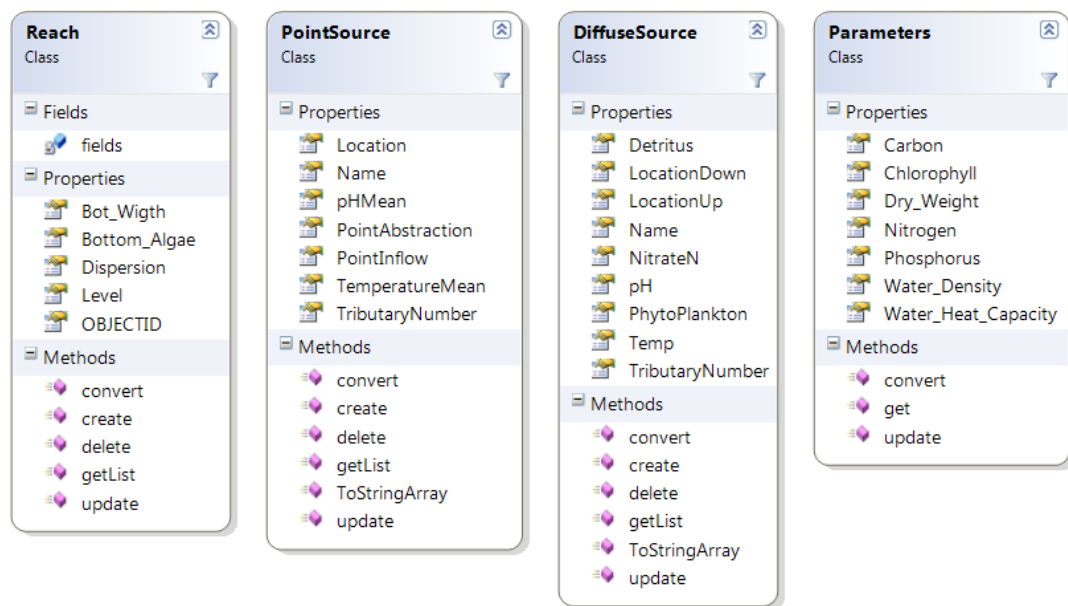


Рисунок 4.10 — Клас `Reach`, `PointSource`, `DiffuseSource`, `Parameters`

Клас `ArcGisTable.cs` відповідає за підключення до гео-бази даних.

Клас `FileWork.cs` є класом, що відповідає за роботу з файлами. А саме, він створює файл, який використовує `Ecogis`, як вхідні дані для опрацювання. Та зчитує інформацію з вихідного файлу.

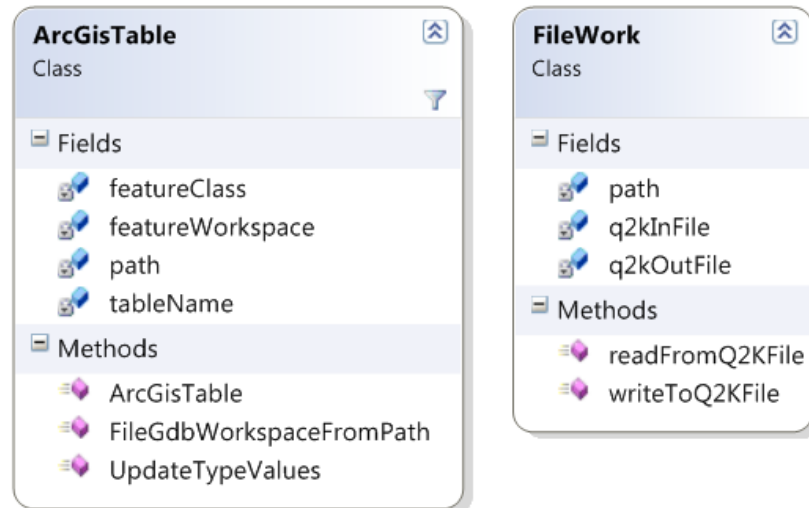


Рисунок 4.11 — Клас ArcGisTable, FileWork

4.3 Опис функціональних особливостей системи

В процесі проектування було використано діаграму прецедентів для відображення відношень між актором та прецедентами.

Діаграма прецедентів є засобом для моделювання системних динамічних аспектів, які у моделюванні поведінки, класу, системи і підсистеми грають основну роль. У кожній такої діаграми безліч акторів, прецедентів і відносин між ними [14].

Велике значення діаграма прецедентів має для специфікації, візуалізації та документування поведінки системи. Використовуючи її, розробнику легше розуміти систему, підсистему або класи, а також поглянути ззовні на переваги використання елементів для того чи іншого контексту. Діаграма представляє особливу важливість для проведення тестування виконуваних систем при прямому проектуванні, а також для кращого розуміння їх внутрішнього устрою, особливо при зворотному проектуванні.

У мові UML визначено такі типи відношення:

- залежність;
- асоціація;

- узагальнення;
- реалізація.

Ці відношення є основними єднальними конструкціями UML і також як сутності застосовуються для побудови моделей [15].

Залежність (Dependency) — це семантичне відношення між двома сутностями, при якому зміна однієї з них, незалежної, може вплинути на семантику іншого, залежного.

Асоціація (Association) — структурне ставлення, що описує сукупність смислових або логічних зв'язків між об'єктами.

Узагальнення (Generalization) — це відношення, при якому об'єкт спеціалізованого елемента (нащадок) може бути підставлений замість об'єкта узагальненого елемента (предка). При цьому, відповідно до принципів об'єктно-орієнтованого програмування, нащадок (child) успадковує структуру і поведінку свого предка (parent).

Реалізація (Realization) є семантичним відношенням між класифікаторами, при якому один класифікатор визначає зобов'язання, а інший гарантує його виконання. Ставлення реалізації зустрічаються у двох випадках:

- між інтерфейсами і реалізують їх класами чи компонентами;
- між прецедентами і реалізують їх кооперації.

При цьому загальні властивості варіантів використання можуть бути представлені різними способами: за допомогою відношень включення, розширення і узагальнення.

Програмна система надає актору “Користувач” такі можливості (рисунки 4.12):

- вводити дані про забруднення на площині;
- вводити дані про точкове забруднення;
- переглядати результат моделювання забруднення;
- перегляд даних аналізу метеостанцій;
- перегляд даних стану атмосфери;

— редагування даних метеостанцій;



Рисунок 4.12 — Функціональна схема

4.4 Висновки до розділу

Система побудована на основі архітектурного шаблону MVC, з використанням усіх його переваг. Його недоліки були уникненні використанням ООП. Структурно було використано десять класів, кожний із яких відповідає за один із аспектів роботи програми.

В цілому розроблений програмний продукт має зручний інтерфейс для ефективної роботи та моделювання забруднень.

5 МЕТОДИКА РОБОТИ КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

Розроблене програмне забезпечення є настільним додатком.

5.1 Інсталяція програмного забезпечення та системні вимоги

Для його роботи на комп'ютері має бути встановлена операційна система Windows 10. Обсяг оперативної пам'яті має бути не меншим 1 Гбайта, на жорсткому диску має бути не менше 20 Мбайтів вільного простору.

Для встановлення програмного забезпечення на комп'ютері досить скопіювати папку з програмою в користувацьку папку.

5.2 Сценарії роботи користувача з системою

Користувач системи під час роботи може використовувати меню, контекстні меню, різноманітні панелі з кнопками. Основні елементи інтерфейсу інтуїтивно зрозумілі.

Після запуску програми з'являється вікно роботи з програмою (рисунок 5.1), в якій користувач може починати роботу з картою та її функціоналом.

Зліва можна побачити панель рівнів(рисунок 5.2), за допомогою якої можна включати чи виключати відображення того чи іншого рівня або змінювати порядок відображення рівнів на карті.

Зверху знаходиться панель елементів за допомогою якої можна працювати з картою та її елементами. Так як програмне забезпечення являється модулем підключення до системи ArcMap то в нас є доступ до його функціоналу.

Кнопки модулю проекту:

— відкрити вікно для введення даних;

— перегляд результатів.

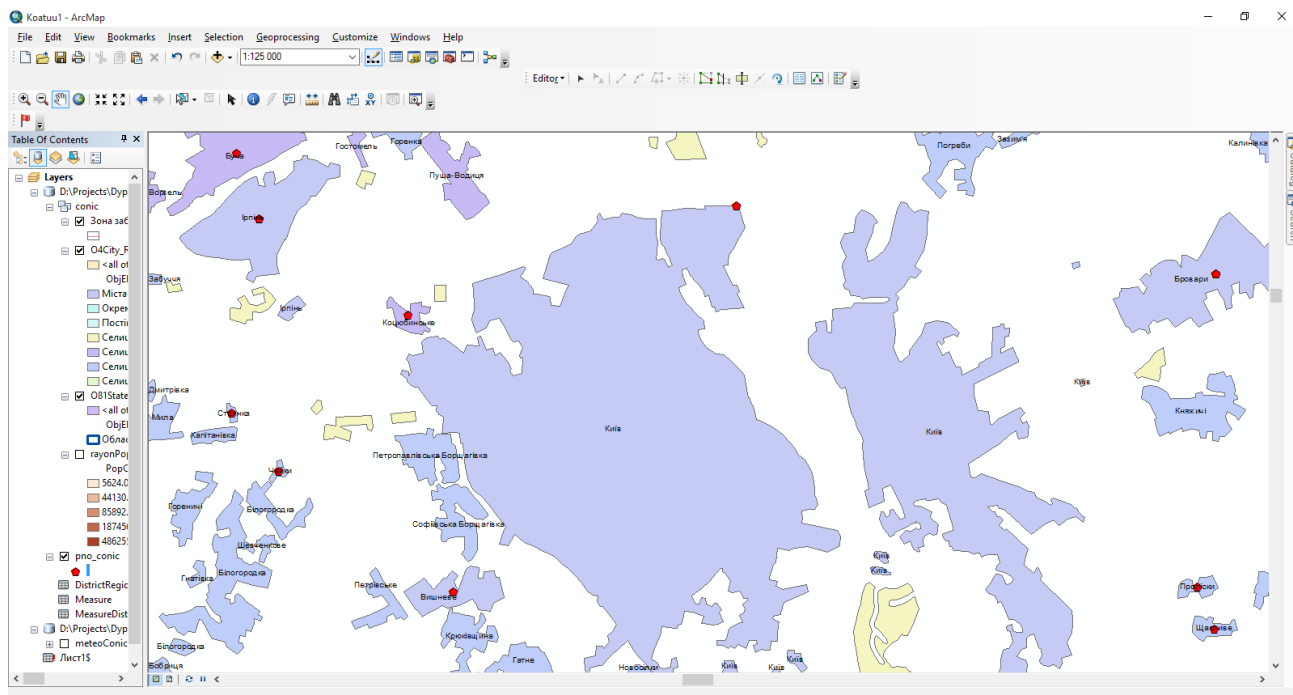


Рисунок 5.1 — Головна форма програми

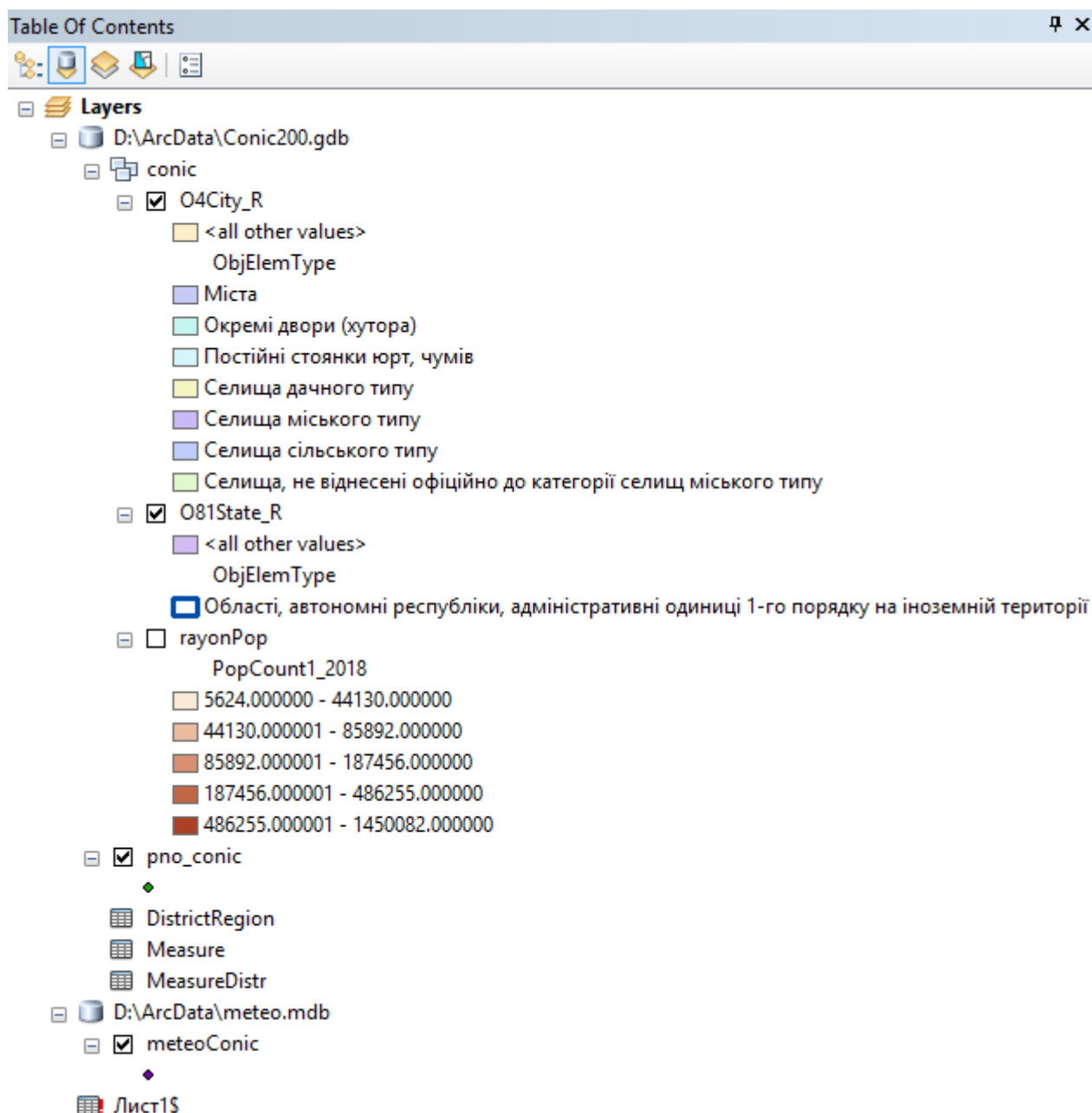


Рисунок 5.2 — Панель рівнів

Якщо користувач натисне на метостанцію він перейде до таблиці в якій відображаються всі дані про аналіз забруднення повітря (рисунок 5.3). У вкладці Parameters знаходяться параметри, коефіцієнти та типи модулів для обчислення, що будуть використовуватися для проведення моделювання (рисунок 5.4).

ObjId	ObjName	containerNames	dangeAgentName	dangeAgentVolume	dangeAgentId
591		2	Хлор рідкий	42	449
591		2	Хлор рідкий	48	449

KodStation	date	KodDirection	WinterMeter	Temperatura	TochkaRosy	Preshure	napr
33345	26.03.2006 12:00	20	2	6,1	-0,5	996,9	195
33345	19.03.2006 18:00	24	4	1,5	-0,7	985,2	235
33345	24.03.2006 6:00	34	3	-3,3	-8,8	995,7	335

Моделювати

Рисунок 5.3 — Вікно аналізу метеостанцій

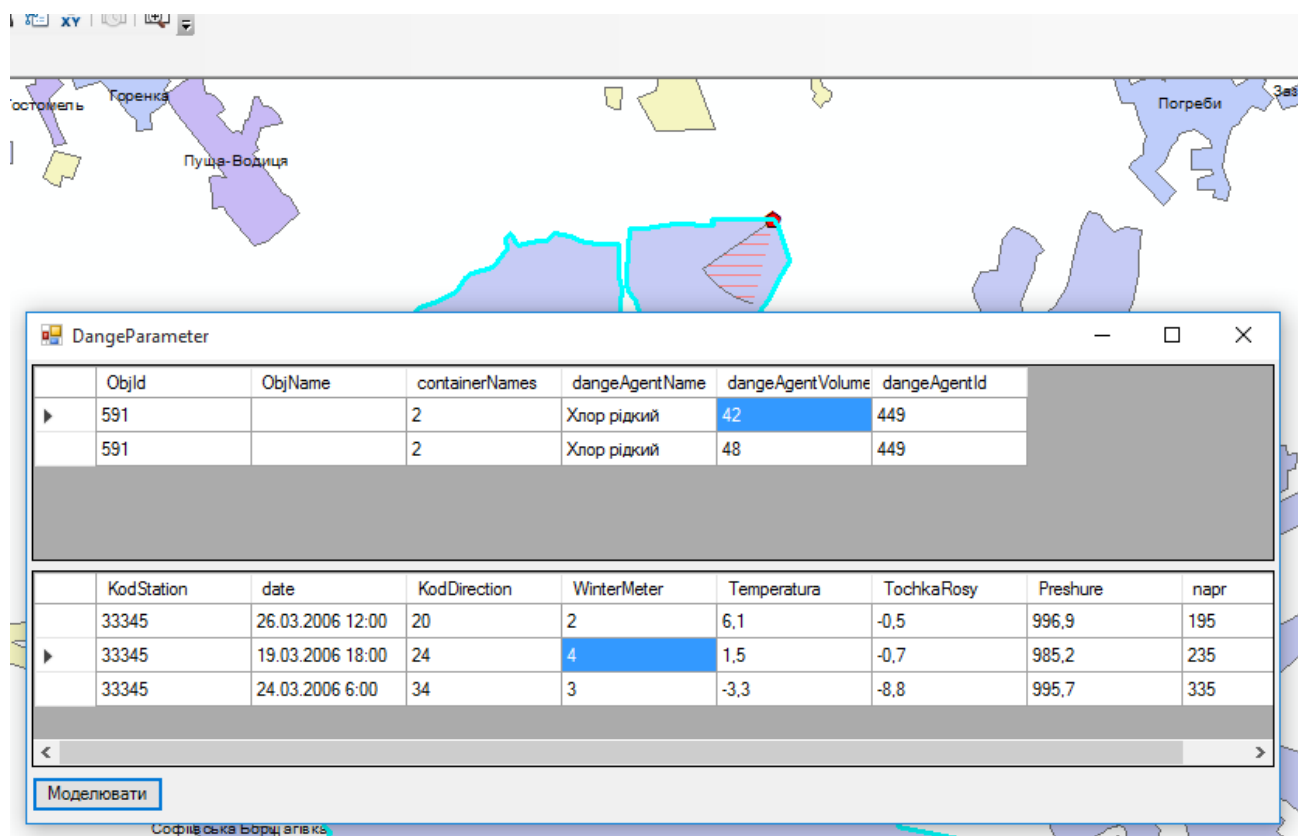


Рисунок 5.4 — Вкладка параметрів

При натисненні на вкладку з назвою Pollutions користувач перейде на вікно з кнопками (рисунок 5.5), а саме Add diffuse source і Add point source, які

викликають діалогові вікна для введення вхідних даних (рисунок 5.4-5.5) відповідно до цих параметрів належать такі властивості:

- місцезнаходження забруднень;
- інтенсивність;
- температура;
- тип забруднення.

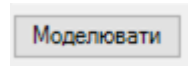


Рисунок 5.5 — Вікно моделювання

Кнопка моделювання викликає запуск моделювання забруднень використовуючи програму EcoGis. Після натиснення програма зчитує дані про забруднення та дані метеостанцій та їх параметри. Перетворює всі вхідні параметри в файл на вхід програмі. Зчитує вихідний результат та перетворює їх в геодані для ArcGis(Рисунок 5.6)

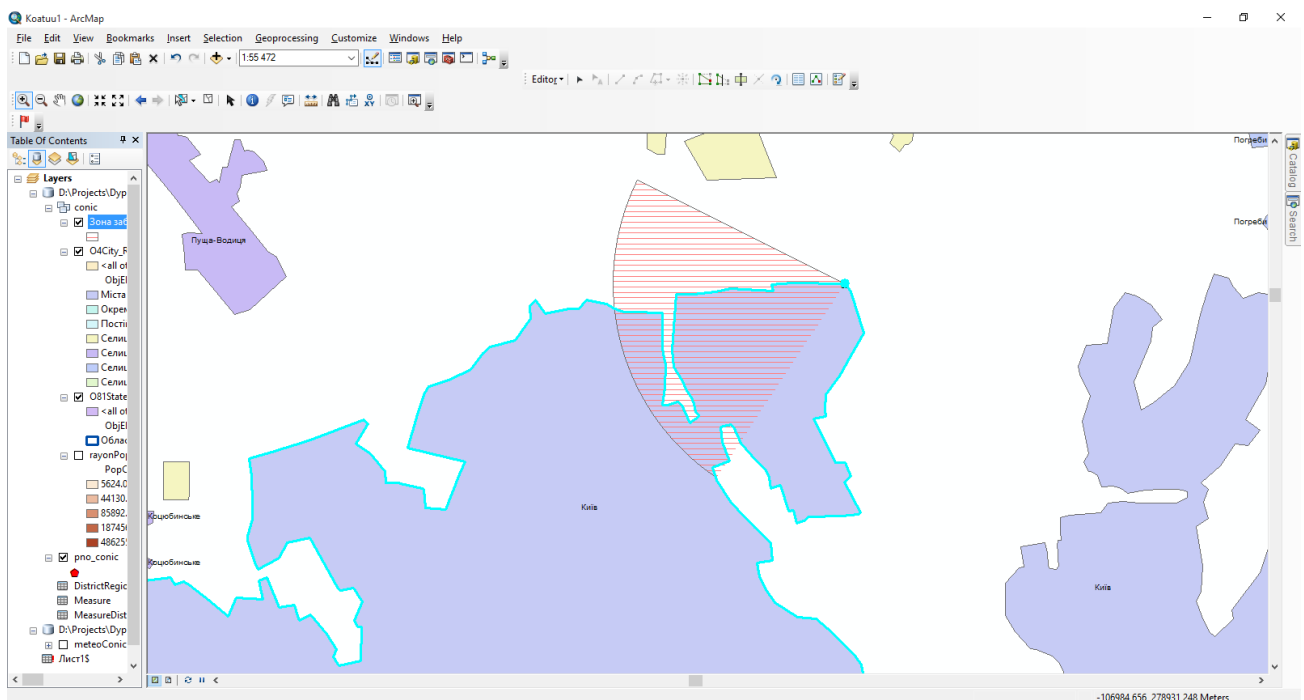


Рисунок 5.6 — Результати моделювання

5.3 Висновки до розділу

У розділі описано принцип роботи з системою. Важливо відмітити, що система складається з двох частин: моделювання карти і забруднення та бази даних метеостанцій і навколишнього середовища. Користувач має можливість взаємодіяти з обома частинами. У розділі перераховані та описані функції, які може виконувати система.

ВИСНОВКИ

Створено програмний комплекс для моделювання хімічного забруднення повітря міста Києва. Створений комплекс реалізує створення моделі даних в просторовій базі даних ArcGis, конвертує та відображає в ArcGis результатів моделювання та надає можливість провести аналіз наслідків розповсюдження забруднювачів.

Система інтегрує дані з геоінформаційної бази даних в яку будуть включені топографічні та тематичні карти , метеостанції та параметри навколишнього середовища.

Під час проходження практики було покращено навички розробки програмних продуктів за допомогою середовища розробки програмного забезпечення Visual Studio та мови програмування C#.

Також були отримані навички застосування архітектури впровадження залежностей та архітектури програмного забезпечення. Було виконано глибоке дослідження ArcObjects SDK. Використання ArcObjects SDK надало можливість відображення карт.

У порівнянні з існуючими аналогами на ринку програмний комплекс має зручний та зрозумілий інтерфейс для користувача, має велику аналітичну потужність.

Таким чином, проходження практики поглибило знання різноманітних технологій розробки клієнтських застосунків та взаємодії зі сторонніми сервісами, використання яких є невід'ємною частиною сучасних програмних продуктів.

.

.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Группа компаний НКФ «Волга» - эксперт в области воздушных ресурсов [Электронный ресурс] — Режим доступа: <http://www.volgaltd.ru/eng/programs/>
2. McDonald M.G. & Harbaugh, A.W. (2003). "The History of MODFLOW". Ground Air. 41 (2): 280–283
3. Нейгел К. С# 4.0 и платформа .NET 4 для профессионалов / К. Нейгел, Б. Ивсен, Дж. Глин, К. Уотсон. — М.: Издательский дом «Вильямс», 2011. — 1440 с.
4. Шилд Г. С# 4.0: полное руководство / Г. Шилдт. — М.: Издательский дом «Вильямс», 2011. — 1056 с.
5. Введение в язык С# и .NET Framework [Электронный ресурс]. — Режим доступа: <https://msdn.microsoft.com/uk-ua/library/z1zx9t92.aspx>
6. Инкапсуляция, полиморфизм, наследование [Электронный ресурс]. — Режим доступа: <http://www.codenet.ru/progr/cpp/ipn.php>
7. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования. / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. — СПб.: Питер, 2007. — 366 с.
8. .NET Framework 4.5 [Электронный ресурс]. — Режим доступа: [https://msdn.microsoft.com/ru-ru/library/dd489454\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/dd489454(v=vs.110).aspx)
9. Introduction to WPF [Электронный ресурс] — Режим доступа: [https://msdn.microsoft.com/en-us/library/aa970268\(v=vs.100\).aspx](https://msdn.microsoft.com/en-us/library/aa970268(v=vs.100).aspx)
10. ArcObjects API Reference for .NET Documentation for the most recent release of ArcObjects [Электронный ресурс] — Режим доступа: http://resources.arcgis.com/en/help/arcobjects-net/componenthelp/index.html#/ArcObjects_namespaces/
11. Калиткин Н. Н. Численные методы. — Главная редакция физико-математической литературы изд-ва «Наука», 1978. — 512 с.

- 12.Как работает метод обратных взвешенных расстояний [Электронный ресурс] — Режим доступа:
<https://desktop.arcgis.com/ru/arcmap/latest/extensions/geostatistical-analyst/how-inverse-distance-weighted-interpolation-works.htm>
- 13.Буч Г., Рамбо Дж., Джекобсон А. Язык UML. Руководство пользователя. — Пер. с англ. — М.: ДМК, 2000. — 432 с.
- 14.Івашин В.В. Автоматизоване проектування систем внутрішнього відеоспостереження / В.В. Івашин // Сучасні аспекти розробки програмного забезпечення: Збірник наукових праць III науково-практичної дистанційної конференції молодих вчених і фахівців з розробки програмного забезпечення, 15 квітня 2016 р. — Черкаси: видавець Чабаненко Ю.А., 2015. — С. 65-68.
- 15.Сучасні проблеми наукового забезпечення енергетики: Матеріали XVI Міжнародної науково-практичної конференції аспірантів, магістрантів і студентів, м. Київ, 24–27 квітня 2018 р. У 2 т. — К. : 7 КПІ ім. Ігоря Сікорського», 2018. — Т. 2. — 298 с.
- 16.Beyer, P. Analysis of Bias and Sources of Variance for Volunteer Air Quality Data, MS thesis, Univ of Iowa, Iowa City, IA. 2008
- 17.Borah, D. K., Yagow, G., Saleh, A., Barnes, P. L., Rosenthal, W., Krug, E. C., and Hauck, L. M. Sediment and nutrient modeling for TMDL development and implementation. T. ASABE, 49(4), 2006, 967-986.
- 18.Salvai, A. Bezdan, A., 2008: Air Quality Model QUAL2K in TMDL Development, BALWOIS 2008, Ohrid, R Macedonia 27- 31th, May 2008.
http://balwois.com/balwois/administration/full_paper/ffp-1161.pdf

ДОДАТОК 1

Моделювання забруднення повітряного середовища міста Києва

Специфікація

УКР.НТУУ “КПІ”. ТЕФ_АПЕПС_ТМ52

Аркушів 1

2019

Позначення	Найменування	Примітки
Документація		
УКР.НТУУ «КПІ ім. І.Сікорського». TM5201_19Б 81-1	Записка_Бичков_TM4 1.docx	Пояснювальна записка
Компоненти		
УКР.НТУУ «КПІ ім. І.Сікорського». TM52_19Б 12-01	Identification/UMO_Signals_ident/Form1.cs	Проведення ідентифікації
УКР.НТУУ «КПІ ім. І.Сікорського». TM52_19Б 12-02	Program.cs	Модулі клієнтської частини програмного продукту
УКР.НТУУ «КПІ ім. І.Сікорського». TM52_19Б 12-03	Form.cs	Модуль інтерфейсу клієнтської частини
УКР.НТУУ «КПІ ім. І.Сікорського». TM52_19Б 12-04	Опис.docx	Опис модуля інтерфейсу клієнтської частини програми
УКР.НТУУ «КПІ ім. І.Сікорського». TM52_19Б 12-05	ModCompDB.mdf	База даних

ДОДАТОК 2

Моделювання забруднення повітряного середовища міста Києва

Текст програми

УКР.НТУУ “КПІ”. ТЕФ_АПЕПС_ТМ52

Аркушів 6

2019

```

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;
using ESRI.ArcGIS.Desktop.AddIns;
using ESRI.ArcGIS.Geometry;
using ESRI.ArcGIS.Geodatabase;
using ESRI.ArcGIS.Carto;
using ESRI.ArcGIS.ArcMapUI;
using ESRI.ArcGIS.DataSourcesGDB;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace RelaitonSelect
{
    public class RelSelect : ESRI.ArcGIS.Desktop.AddIns.Tool
    {
        public List<MeteoStation> meteoStations = new List<MeteoStation>();
        public RelSelect()
        {
            //http://desktop.arcgis.com/ru/arcmap/10.3/manage-data/geometric-networks/what-are-geometric-networks-.htm
            protected override void OnMouseDown(MouseEventArgs arg)
            {
                //if (arg.Button == System.Windows.Forms.MouseButtons.Left) // check for left
                mouse button if you like
                // convert point from 'screen' to 'map' coordinates
                IPoint MouseMapPoint = (ArcMap.Document.FocusMap as
                IActiveView).ScreenDisplay.DisplayTransformation.ToMapPoint(arg.X, arg.Y);
                // clear selection first
                ArcMap.Document.FocusMap.ClearSelection();
                // Select using the shape (point) to
                // select the feature(s) - false to select any intersecting, true to select just the first
                ArcMap.Document.FocusMap.SelectByShape(MouseMapPoint,
                (ArcMap.Application as IMxApplication).SelectionEnvironment, false);

                IEnumFeature EnumFeatures =
                (IEnumFeature)ArcMap.Document.FocusMap.FeatureSelection;
                //!!!Треба використати наступні два оператори
                IEnumFeatureSetup efs = EnumFeatures as IEnumFeatureSetup;

```

```

    efs.AllFields = true;
    //!!!

    IFeature ThisFeature = EnumFeatures.Next();
    //ThisFeature.

do
{
    // get the geometry
    IGeometry ThisGeom = ThisFeature.ShapeCopy; // IMPORTANT! USE
SHAPECOPY not SHAPE
                                // do something different for each geometry type
    if (ThisGeom.GeometryType == esriGeometryType.esriGeometryPoint)
    {
        // something for points (note: does not include multipoint)
    }
    else if (ThisGeom.GeometryType == esriGeometryType.esriGeometryPolygon)
    {
        // something for polygons

    }
    else if (ThisGeom.GeometryType == esriGeometryType.esriGeometryPolyline)
    {
        // something for polylines (note: polylines are different to lines)
    }
    else
    {
        // unrecognized geometry type
    }
    ITopologicalOperator topologicalOperator = MouseMapPoint as
ITopologicalOperator;
    IPolygon polygon = topologicalOperator.Buffer(1000) as IPolygon;
    ArcMap.Document.FocusMap.ClearSelection();
    // Select using the shape (point) to
    // select the feature(s) - false to select any intersecting, true to select just the first
    ArcMap.Document.FocusMap.SelectByShape(polygon, (ArcMap.Application as
IMxApplication).SelectionEnvironment, false);
    (ArcMap.Document.FocusMap as IActiveView).Refresh();
    // IGeometry bufForVES= ThisGeom.
    IEnumFeature EnumFeaturesVes =
(IEnumFeature)ArcMap.Document.FocusMap.FeatureSelection;
    IEnumFeatureSetup efsVES = EnumFeatures as IEnumFeatureSetup;
    efsVES.AllFields = true;
    IFeature featVES = EnumFeaturesVes.Next();

```

```

int fldVESId = featVES.Fields.FindField("REESTR_ID");
do
{
    // get the geometry
    IGeometry ThisGeomVES = featVES.ShapeCopy; // IMPORTANT! USE
SHAPECOPY not SHAPE

                                // do something different for each geometry type
    if (ThisGeomVES.GeometryType == esriGeometryType.esriGeometryPoint)
    {
        // something for points (note: does not include multipoint)
        if (featVES.Class.AliasName == "pno_conic")
        {
            List<Dange> danges = new List<Dange>();
            int _kodStation;
            int _id = (int)featVES.Value[fldVESId];
            // MessageBox.Show(_id.ToString());
            SqlConnection mySqlConnection = new SqlConnection(@"Data
Source=DESKTOP-DNKCRQV;Initial Catalog=EkoGis;Integrated Security=SSPI;");
            mySqlConnection.Open();
            using (SqlCommand command = new SqlCommand(@"SELECT
pno1.idobj, containers._name_, containers._dangclass_,

containers._dangagent_,containers._volume_,pno1.KodStation,containers._idpollagent_
FROM pno1 INNER JOIN containers
                ON pno1.idobj = containers._idobj_
                WHERE      (pno1.idobj = " + _id.ToString()+") AND
(pno1.idchemcat = 3)", mySqlConnection))

                using (SqlDataReader reader = command.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        Dange _dange = new Dange();
                        _dange.ObjId = (int)reader.GetInt32(0);
                        _dange.ObjName = ""; // reader.GetString(1);
                        _dange.containerNames = reader.GetString(1);
                        _dange.containerNames = reader.GetString(2);
                        _dange.dangeAgentName = reader.GetString(3);

                        _dange.dangeAgentVolume = reader.GetDouble(4);
                        _dange.dangeAgentId = reader.GetInt32(6);
                        _kodStation = reader.GetInt32(5);
                        danges.Add(_dange);
                    }
                }
            }
        }
    }
}

```

```

    }
    // List<MeteoStation> meteoStations = new List<MeteoStation>();
    using (SqlCommand command = new SqlCommand(@"SELECT
mt.[date] ,mt.[KodStation],mt.[KodDirection]

,mt.[WinterMeter],mt.[Temperatura],mt.[TochkaRosy],mt.[Preshure],mt.[PrecipMesure],v
d.napr

FROM [EkoGis].[dbo].[Meteo] as mt join
[EkoGis].[dbo].[vind_directions1] as vd on vd.code_wind_direction=mt.[KodDirection]
where KodStation=33345", mySqlConnection))

    using (SqlDataReader reader = command.ExecuteReader())
    {
        while (reader.Read())
        {
            MeteoStation _meteo = new MeteoStation();
            _meteo.date = reader.GetDateTime(0);
            _meteo.KodStation = reader.GetInt32(1);
            _meteo.KodDirection = reader.GetInt32(2);
            _meteo.WinterMeter = reader.GetInt32(3);
            _meteo.Temperatura = reader.GetDouble(4);
            _meteo.TochkaRosy = reader.GetDouble(5);
            _meteo.Preshure = reader.GetDouble(6);
            _meteo.napr = reader.GetInt32(8);
            // _meteo.PrecipMesure = reader.GetInt32(7);
            meteoStations.Add(_meteo);
        }
    }
    DangeParameter dangeParameter = new
RelaitonSelect.DangeParameter(danges, meteoStations, featVES);
    dangeParameter.Show();
    dangeParameter.mMap= ArcMap.Document.FocusMap;
}

}
else
{

}

featVES = EnumFeaturesVes.Next();
} while (featVES != null);
ThisFeature = EnumFeatures.Next();
} while (ThisFeature != null);
base.OnMouseDown(arg);

```



```

    }
    protected override void OnUpdate()
    {
        Enabled = ArcMap.Application != null;
    }

//http://resources.esri.com/help/9.3/arcgisengine/ArcObjects/esrigeodatabase/IRelationship
_Example.htm
    public void IRelationship_Example(IFeature feature)
    {
        //assumes that only one RelationshipClass exists for the Origin feature class
        //IWorkspaceFactory wwf = new FileGDBWorkspaceFactory();
        //IFeatureWorkspace featWS =
wwf.OpenFromFile(@"D:\Викладання\GenPlan\GenPlan.gdb", 0) as IFeatureWorkspace;

        IEnumRelationshipClass enumRelClass =
feature.Class.get_RelationshipClasses(esriRelRole.esriRelRoleAny);
        IRelationshipClass relClass = enumRelClass.Next();
        //if a feature with no Relationships established has been selected, exit
        if (relClass == null)
        {
            return;
        }
        ESRI.ArcGIS.esriSystem.ISet relSet =
relClass.GetObjectsRelatedToObject((IObject)feature);
        relSet.Reset();

        IFeature destinationFeature = (IFeature)relSet.Next();
        int fld_diameter = destinationFeature.Fields.FindField("d");

        while (destinationFeature != null)
        {
            int _nom = destinationFeature.OID;//номер труби
            double diameter = destinationFeature.Value[fld_diameter];//діаметр труби
            MessageBox.Show("Номер труби=" + _nom.ToString() + " діаметр=" +
diameter.ToString());
            destinationFeature = (IFeature)relSet.Next();
        }
    }

}
public class Region
{
    public int Id { get; set; }

```

```

    public string Name{ get; set; }
    public string koatuu { get; set; }
}
public class Dange
{
    public int ObjId { get; set; }

    public string ObjName { get; set; }
    public string containerNames { get; set; }
    public string dangeAgentName { get; set; }
    public double dangeAgentVolume { get; set; }
    public int dangeAgentId { get; set; }
}
public class MeteoStation
{

    public int KodStation { get; set; }
    public DateTime date { get; set; }
    public int KodDirection { get; set; }
    public int WinterMeter { get; set; }
    public double Temperatura { get; set; }
    public double TochkaRosy { get; set; }
    public double Preshure { get; set; }
    public int napr { get; set; }
    // public int PrecipMesure { get; set; }
}
}

```

ДОДАТОК 3

Моделювання забруднення повітряного середовища міста Києва

Опис програми

УКР.НТУУ “КПІ”. ТЕФ_АПЕПС_ТМ52

Аркушів 8

2019

АНОТАЦІЯ

Додаток надає інформацію про програмну систему моделювання хімічного забруднення повітряного середовища міста Києва

Розроблене програмне забезпечення дозволяє отримувати модель хімічного забруднення атмосфери міста Києва, виконувати попередньо написані запити, отримувати та виводити результат.

Клієнтський застосунок було розроблено в середовищі Microsoft Visual Studio 2013 з використанням платформи .NET та мови C# на базі геінформаційної системи ArcGis.

ЗМІСТ

1. ЗАГАЛЬНІ ВІДОМОСТІ	70
2. ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ.....	71
3. ОПИС ЛОГІЧНОЇ СТРУКТУРИ	72
4. ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ.....	73
5. ВИКЛИК І ЗАВАНТАЖЕННЯ.....	74
6. ВИХІДНІ ДАНІ	75
7. ВХІДНІ ДАНІ	76

1 ЗАГАЛЬНІ ВІДОМОСТІ

Серверна частина програмного продукту "Моделювання забруднення повітряного середовища міста Києва" є десктопним додатком, написаним мовою програмування C#.

Клієнтський інтерфейс написаний мовою програмування C# з використанням геоінформаційної бази ArcGis та платформою .NET.

Для функціонування системи необхідний комп'ютер з операційною системою Windows (7 та вище) та система керування базами даних MS SQL Server .

2 ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Система "Моделювання забруднення повітряного середовища міста Києва" призначена для науковців, які проводять різноманітні спостереження в атмосфері. Даний застосунок також може бути корисним в практичній сфері державного управління, бізнесу, науки, освіти та ЗМІ, адже ідентифікація забруднень в атмосфері є досить важливою. Не менш важливою складовою є те, що даний застосунок був розроблений з метою зробити його користування для користувача якомога простішим та зручнішим.

3 ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Система "Моделювання забруднення повітряного середовища міста Києва" працює методом аналітичного пошуку. Тобто, програма проводить аналіз даних з метеостанцій про стан забруднення атмосфери, інформація про які зберігається в базі даних, імітує екстренну ситуацію та моделює на карті стан хімічного забруднення . Також реалізовано можливість встановити кількість та ідентифікувати тип забруднювачів. Це може допомогти у разі, коли в базі даних не зберігається інформація про тип забруднювачів.

4 ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ

Для функціонування системи "Моделювання забруднення повітряного середовища міста Києва" необхідний будь-який комп'ютер з ОС Windows.

5 ВИКЛИК І ЗАВАНТАЖЕННЯ

Для встановлення програми на новий комп'ютер необхідно (приклад для ОС Windows 10):

- скопіювати папку з програмою на комп'ютер;
- встановити MS SQL Server;
- встановити ArcGis;
- запустити програмний продукт через MS Visual Studio 2013.

6 ВИХІДНІ ДАНІ

Вихідною інформацією є:

- візуалізований список забруднювачів в повітряному середовищі;
- моделювання забруднювачів у атмосфері Києва.

7 ВХІДНІ ДАНІ

Вхідною інформацією є:

- аналіз хімічних забруднювачів з метеостанцій;
- тип забруднювачів у атмосфері Києва.